



PARTICIPANT GUIDE

API MANAGEMENT WITH WEBMETHODS PLATFORM

456-71E

Software AG
Internal Use Only!

This publication is protected by international copyright law. All rights reserved. No part of this publication may be reproduced, translated, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of Software AG.

Software AG and all Software AG products are either trademarks or registered trademarks of Software AG. Other product or company names mentioned herein may be the trademarks of their respective owners.



API Management with webMethods Platform

456-71E

Notes:

Copyright

- This publication is protected by international copyright law. All rights reserved. No part of this publication may be reproduced, translated, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of Software AG.
- Software AG and all Software AG products are either trademarks or registered trademarks of Software AG. Other product or company names mentioned herein may be the trademarks of their respective owners.

Software AG Training | Page 2

Notes:



Welcome to Software AG Training!

Housekeeping items

- Class hours / Refreshments
- Restrooms / Smoking
- Emergency exits
- Sign-in sheets

For everyone's benefit, please:

- Turn off/mute phones
- Check e-mail only at breaks
- Refrain from side discussions during the presentation. It can be very distracting.
- Feel free to ask questions during the lecture.

Software AG Training | Page 3

Notes:

Table of Contents

- ▶ 1. API Management Overview
- ▶ 2. API Gateway Overview
- ▶ 3. API Creation in API Gateway
- ▶ 4. Identification Management
- ▶ 5. Consumer Management
- ▶ 6. Policy Management
- ▶ 7. Traffic Management
- ▶ 8. Routing and Mediation
- ▶ 9. Analytics in API Gateway
- ▶ 10. API Portal Overview
- ▶ 11. API Portal as Provider

Notes:

Table of Contents

- ▶ 12. API Portal as Consumer
- ▶ 13. Analytics in API Portal
- ▶ 14. Packages and Plans
- ▶ 15. Advanced Security
- ▶ 16. API Gateway Administration
- ▶ 17. Customization in API Portal
- ▶ 18. API Portal Administration
- ▶ 19. Wrap Up

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



1

API Management Overview

Notes:

Objectives

At the end of this chapter you ...

- Know about APIs
- Know why we need API Management
- Understand the key features of API Management tools

Notes:

Chapter Contents

- API Management Platform
- API Gateway

Notes:

We Live in an API-Connected World

The diagram illustrates the interconnected nature of various digital ecosystems through APIs. At the center is a central node representing users (two people). Surrounding this central node are several other nodes, each connected by double-headed arrows, indicating bidirectional communication or data exchange:

- Mobile**: Represented by a smartphone icon.
- SaaS**: Represented by a cloud icon.
- Web Apps**: Represented by a computer monitor icon.
- Partners**: Represented by a building icon.
- Things**: Represented by a truck icon.
- Big Data**: Represented by an elephant icon.
- Apps/ Data**: Represented by a database icon with a gear.

Arrows indicate the flow of data or services between these components, showing how APIs facilitate connectivity across different domains such as mobile, web, enterprise, and data management.

- APIs are a foundation of digital transformation:
 - Enable mobile apps, create digital ecosystems across customers and partners

Software AG Training | 1 - 4

We now live in an API-connected world.

APIs have become the standard way of connecting applications, data and devices, and providing services to partners and end users.

APIs are what makes possible the exposing of data and services in web apps, mobile apps, and other connected devices.

APIs provide new models for doing business.

With APIs, the Internet is now a platform for your business, no matter what kind of business you are in.

APIs are not new. But they used to be internal.

Now with common standards and protocols (REST/JSON/etc.), APIs are easy for everyone to adopt.

More and more business will be conducted over the Internet via APIs

Existing technologies, such as Integration, beginning to include API Management, to help define app logic, etc.

APIs Enable Digital Business

- APIs allow connecting applications or computer systems in a standardized and flexible way
- APIs enable exposing data functionality in web apps, mobile apps, and other connected devices
- APIs provide new models for doing business
- APIs transform the Internet into a platform for your business



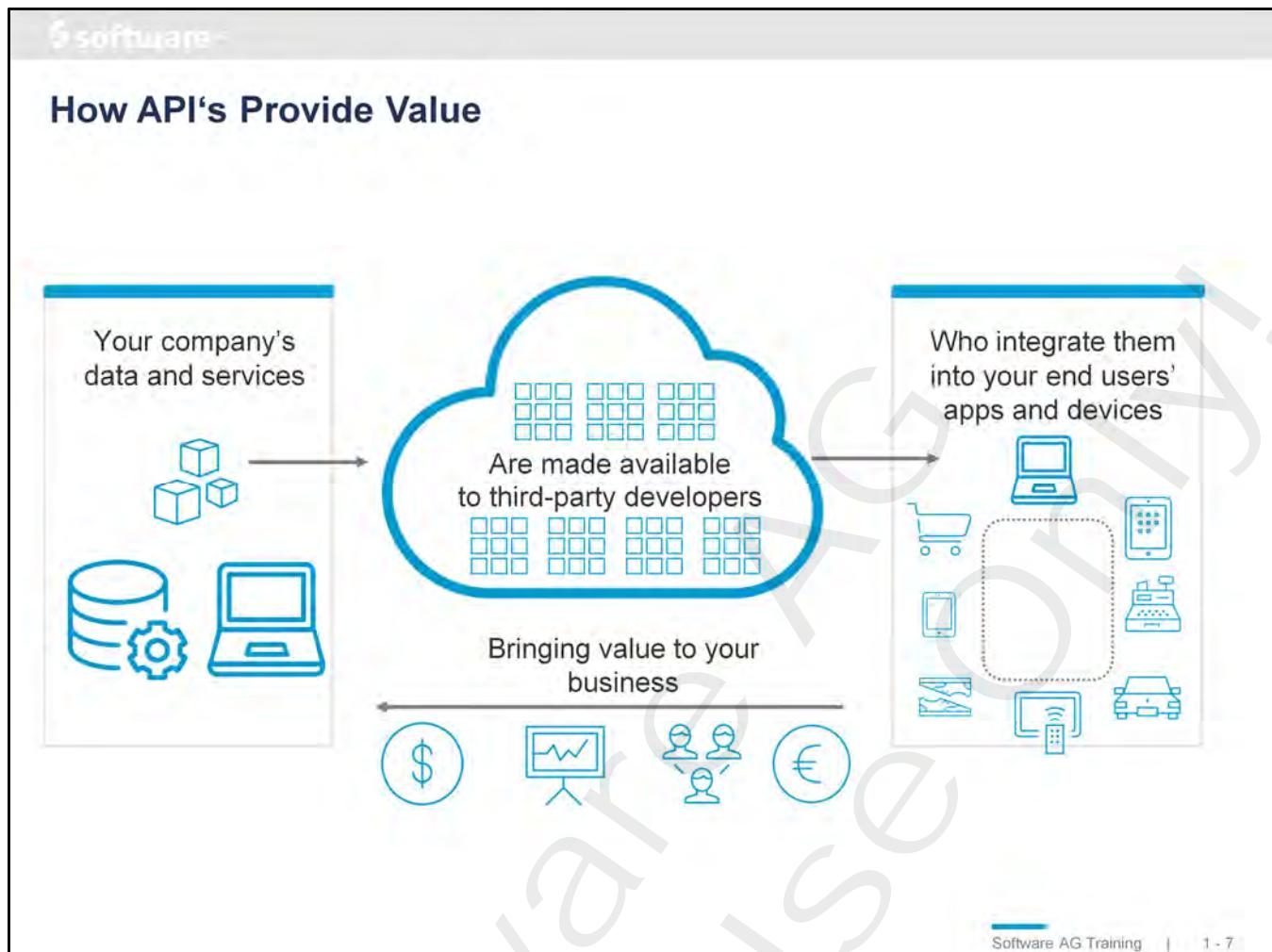
Notes:

Sample Industry Use Cases

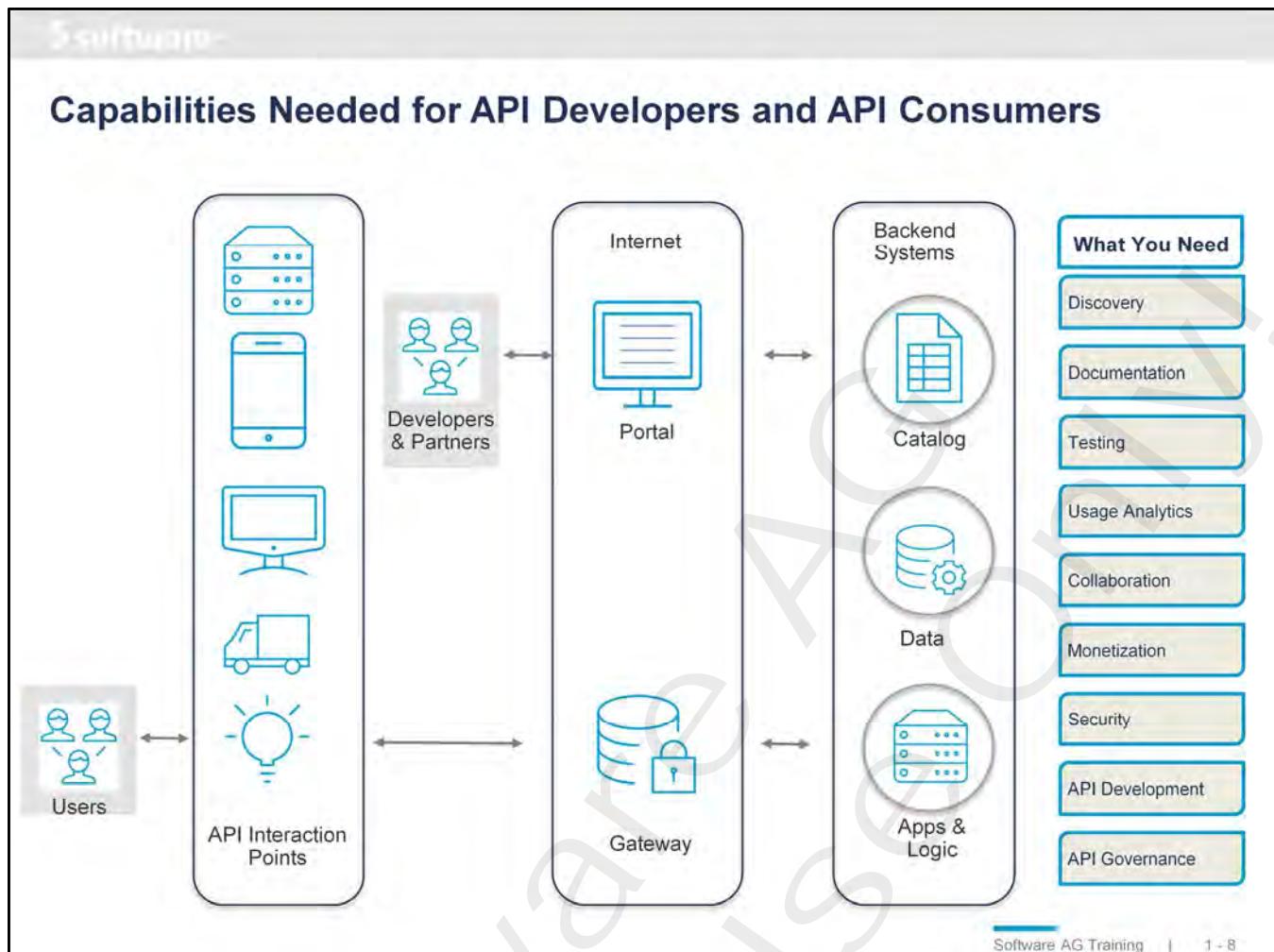
 Banking	Loyalty programs, mobile banking, payment services	 Government	Information transparency, citizen outreach, open data initiatives
 Retail	Omni-channel integration, affiliate programs	 Media	Content subscription, content syndication, customer outreach
 Energy/ Utilities	Home energy management, smart metering	 Telco	Call/messaging integration
 Manufacturing	Partner onboarding	 Automotive	Connected Car Initiatives, Location based Services, Car Remote Control

Software AG Training | 1 - 6

Notes:



Notes:



Notes:

software AG

API's are All about Experience

- Ensuring the quality of the APIs

The diagram illustrates the API development process. It starts with a monitor icon representing a "great Developer Portal experience...". A large blue arrow points from this to a stack of four boxes labeled "REST", "SOAP", "OData", and "Web Sockets". Finally, another blue arrow points to a cluster of icons representing "good APIs!", including three cubes, two wrenches, and two gears.

Experience begins with a great Developer Portal experience...

But just the beginning.

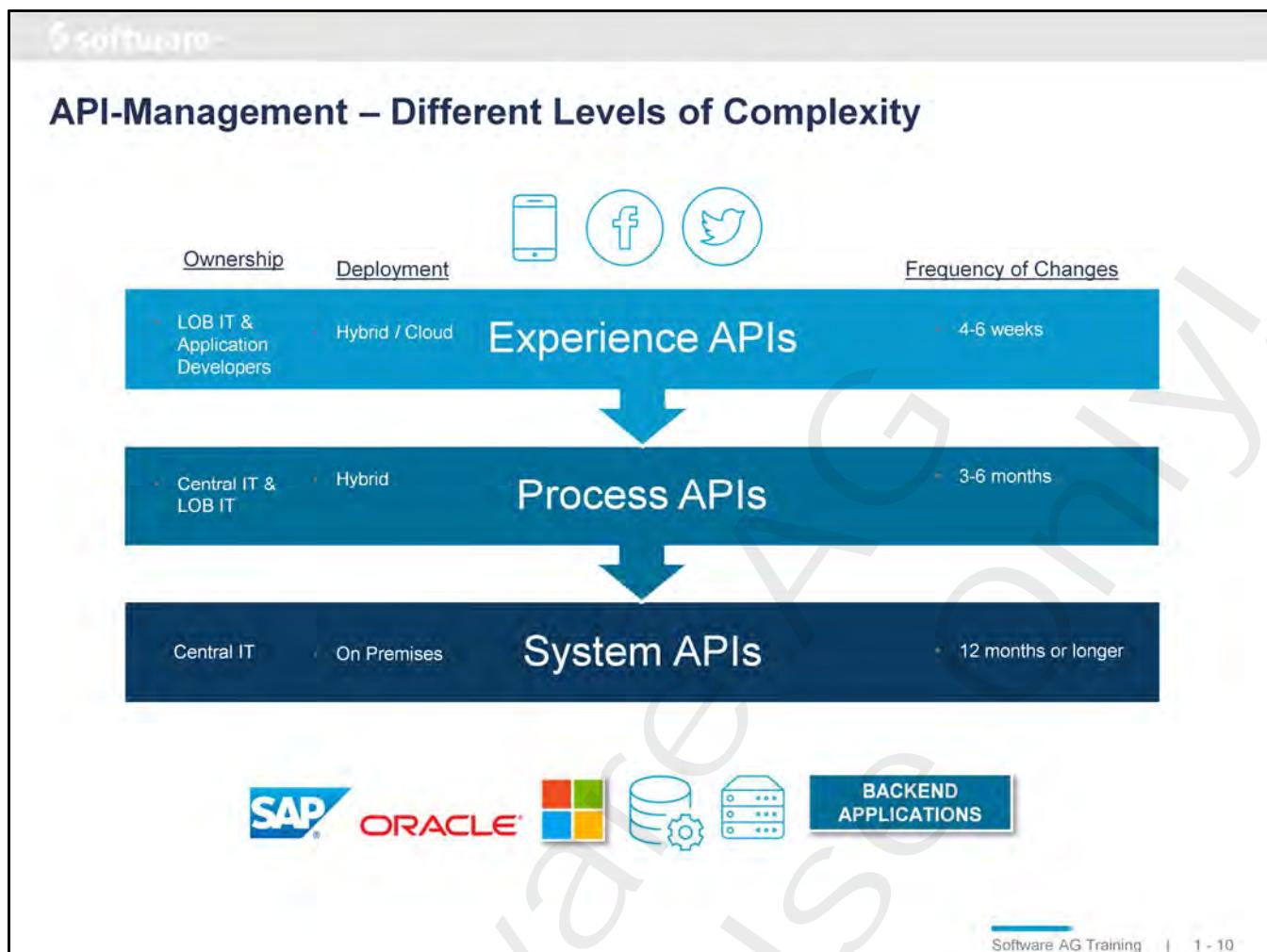
REST
SOAP
OData
Web Sockets

What really matters are good APIs!

Which requires the full spectrum of assets and tools to build your APIs.

Software AG Training | 1 / 9

Notes:



Integration and API Management are converging.

APIs are increasingly being used for B2B transaction exchange

Early adopters are re-thinking API strategy

Pure play API Portal vendors are evolving and broadening portfolio

Systematic approach to API design.

First, we see a systematic approach to API design that helps to ensure the right APIs are being developed, with the right level of developer experience.

Historically, many APIs were like the System APIs presented here.

- Central IT, long term

More commonly we are now seeing higher level APIs that encompass a broader business process.

- Shorter term projects, often driven from departments

Ultimately, we are seeing need for more rapid, higher level customer/developer experience

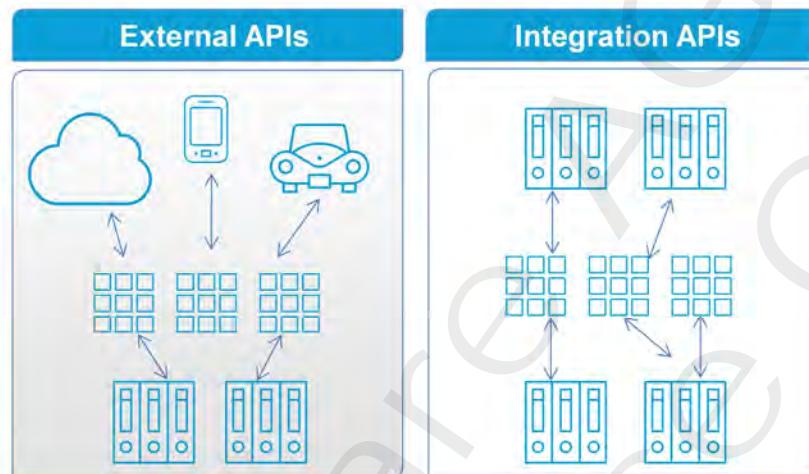
- These might be driven by departments and are geared for improved customer experience

The takeaway is that developers cannot treat all APIs as the same, and need to apply the right level of complexity based on the use case.

Perhaps more importantly though is the need for broader capability support in API Management platforms...

Services / APIs as Building Blocks

- A Service/API approach provides the agility needed for a rapidly changing application landscape
 - Universal reuse of business in cloud, mobile and enterprise
 - Technology agnostic
 - Simple composition of apps



Software AG Training | 1 - 11

Notes:

API Management and Integration

The diagram illustrates the convergence of Integration and API Management. It starts with two boxes labeled 'Integration' and 'API Management' followed by a plus sign. A bracket groups these with three blue callout boxes. The first box contains the text 'Integration (including B2B) and API Management are converging'. The second box contains 'Adoption of new API standards like Swagger, OData, OpenID Connect etc. in addition to REST/JSON'. The third box contains 'Quick API creation tools for faster delivery of digital applications'. Below these three boxes are two ovals: one labeled 'API Standards' and another labeled 'Digital Business APIs', which are also grouped by a bracket.

- Integration (including B2B) and API Management are converging
- Adoption of new API standards like Swagger, OData, OpenID Connect etc. in addition to REST/JSON
- Quick API creation tools for faster delivery of digital applications

Software AG Training | 1 - 12

With the breadth of API use cases we talked about earlier, we see that API development will require more tools.

They will need the full spectrum of capabilities.

There is a degree to which APIs and Integration platforms are converging.

Early adopters are re-thinking API strategy

Pure play API Portal vendors are evolving and broadening portfolio

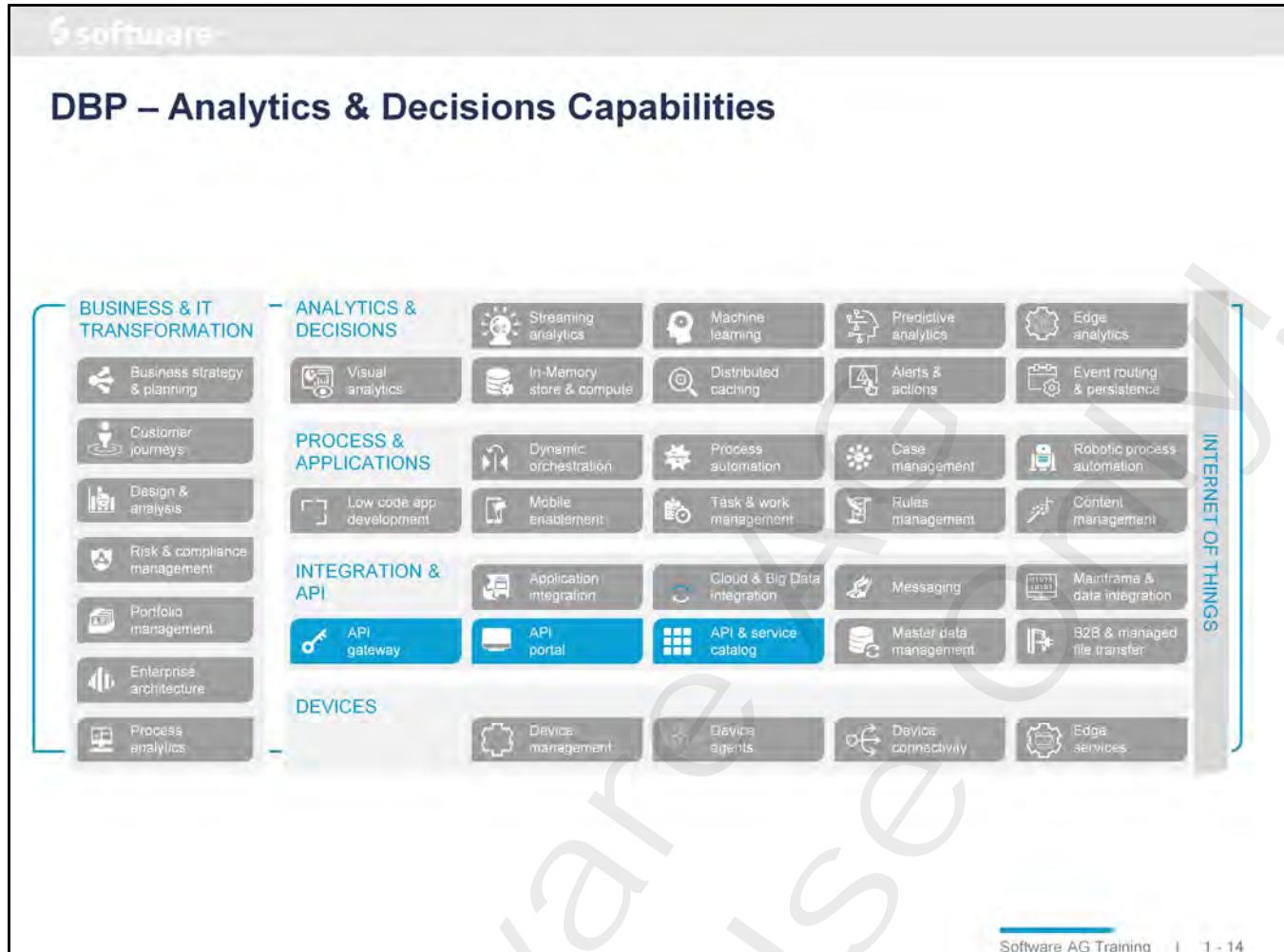
Vendors are launching quick API creating tools for faster delivery

APIs are increasingly being used for B2B transaction exchange



Software AG Training | 1 - 13

Notes:



Notes:



API Management Platform

Notes:

Software AG's API Management Platform

The diagram illustrates the Software AG API Management Platform architecture. It features four main components arranged horizontally:

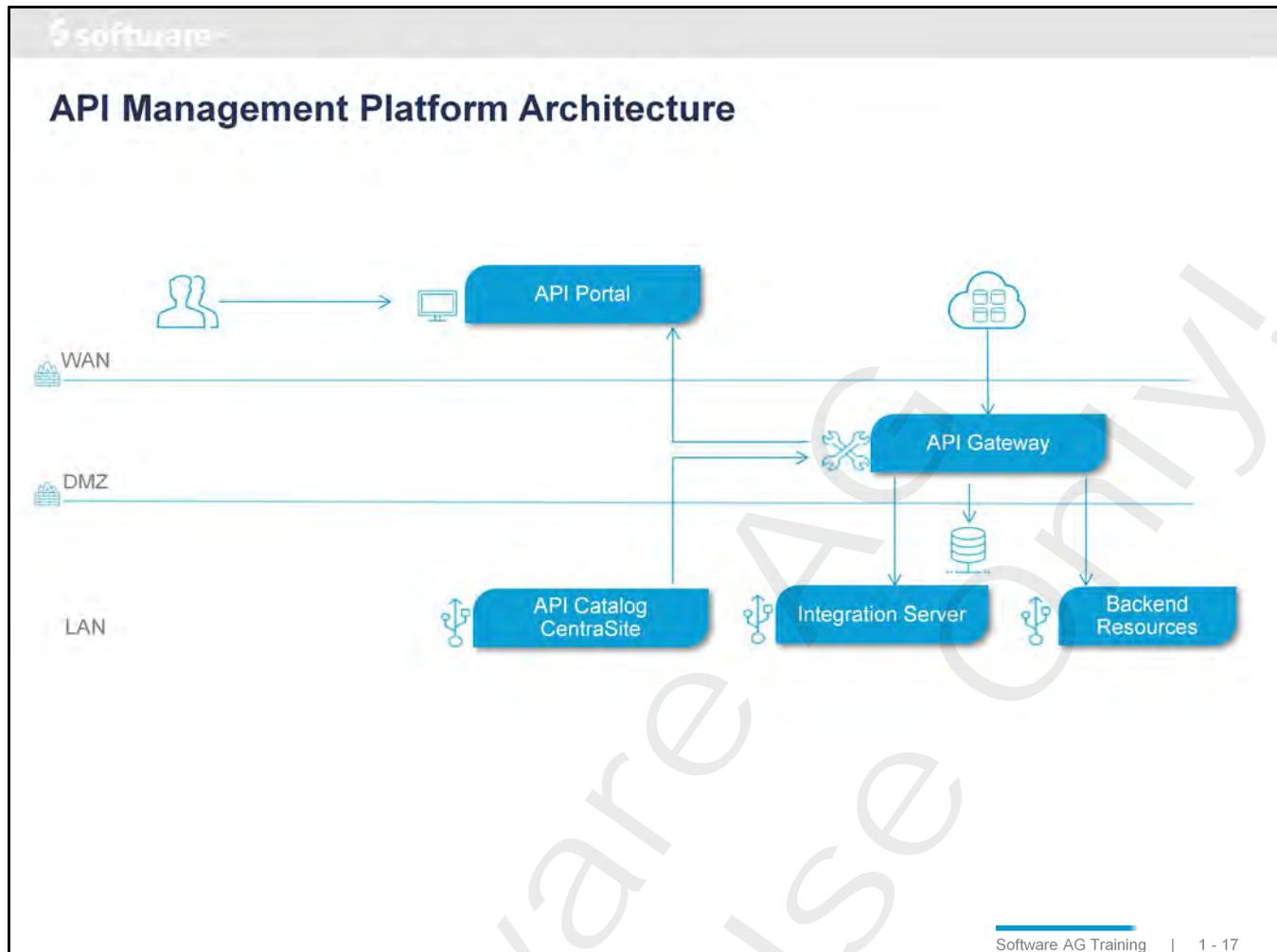
- API Catalog**: Represented by a square icon with three vertical bars and three circles. The features listed are:
 - Define your APIs
 - Discover existing API assets
 - Manage lifecycles
 - Apply runtime governance policies
 - Track dependencies
- API Gateway**: Represented by a key icon. The features listed are:
 - Protect your APIs from unregistered usage
 - Protect your backend systems from malicious attacks
 - Monitor and track usage for monetization and product improvement
- API Portal**: Represented by a monitor icon. The features listed are:
 - Promote APIs
 - Document usage
 - Enable registration
 - Community and collaboration
 - Testing
- Central Cloud Icon**: Represented by a cloud icon with three smaller icons below it: Cloud, Hybrid, and On-Premise.

API Management Platform enables

- Planning, design, implementation, publication, operation, consumption, maintenance and retirement of APIs

Software AG Training | 1 - 16

Notes:



Software AG Training | 1 - 17

Notes:

API Governance: CentraSite

- Build a comprehensive catalog of internal and external APIs
 - Understand dependencies
 - Document your APIs for Developers
 - Classify your APIs
- Manage the API Lifecycle
 - Create Lifecycle Models
 - Create Design Time Policies
 - Consumer Management
- Analyze API
 - Graphical Impact Analysis and dynamic discovery
 - Use Dashboards
 - Run Reports

WAN

DMZ

LAN

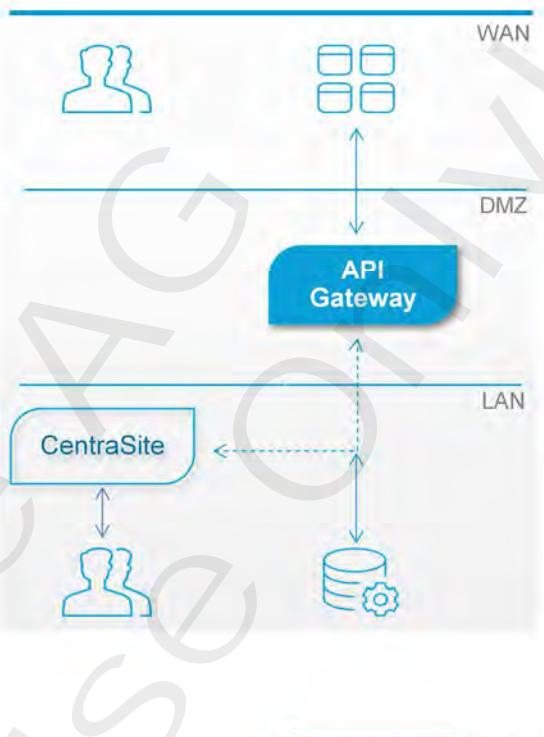
CentraSite

Software AG Training | 1 - 18

Notes:

API Gateway

- API mediation and virtualization
 - Enforce Security, Traffic Management, Monitoring, SLA Management for APIs
 - Intelligent Routing and Load Balancing of Requests
 - Collection of analytical data on API consumption and policy evaluation
- DMZ level protection
 - DoS protection based on IP/Messaging volume
 - IP Blacklisting and Filtering
 - Virus Scanner Integration
 - Reverse Invoke Technology
- Analyze API runtime usage
 - Dashboards
 - Reports and traffic profiles

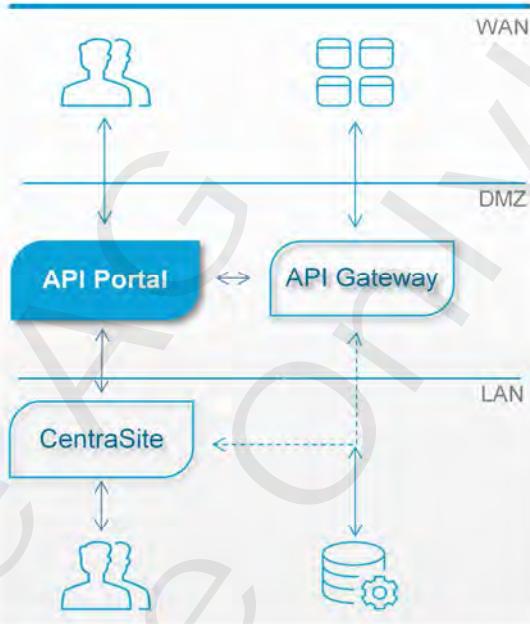


Software AG Training | 1 - 19

Notes:

API Portal

- Self-service portal to expose APIs for developers and B2B partners
- Build-in Usage Analytics
- REST and SOAP API testing
- Metadata driven API Documentation
- Integrated Collaboration
- Multi-Tenant Architecture



Software AG Training | 1 - 20

Notes:



API Gateway

Notes:

API Management



- Decouple consumer and provider
 - Shield consumer from backend service
 - Apply operational changes more easily
 - Simpler onboarding of consumers with distinct requirements
- Centrally configure & enforce Policies
 - Less errors through centralized policies
 - Holistic picture of all enforced policies
 - Easier rollout and scaling
- Separate Business Logic and Integration Logic
 - Adopt changes in integration logic faster
 - Higher stability of business logic layer

Software AG Training | 1 - 22

API Management platform enables enterprises to selectively externalize their new and existing assets as APIs across various channels, monitor the interfaces lifecycle with an integrated infrastructure and make sure the needs of developers and applications using the API are met.

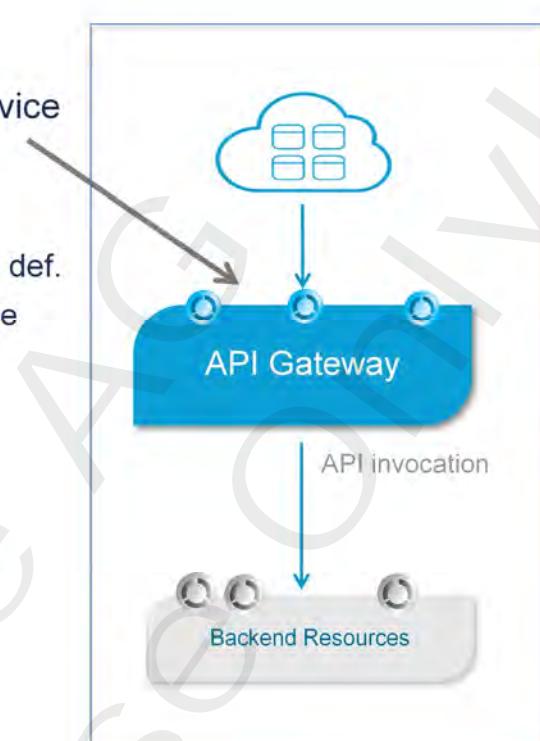
APIs are the new distribution channel for Assets.

Using the integrated infrastructure you can:

- Securely expose your APIs to external developers and partners. A partner is any entity with which your enterprise interacts, such as suppliers and other vendors, dealers, and distributors, customers, government agencies, trade organizations and so forth.
 - You can provide design and runtime governance to the API's
- API Management enables developers, architects and business developers to
- Publish APIs into their organization's central registry
 - Discover APIs and use them to assemble new applications
 - Obtain access to detailed information of the API like list of consumers, technical support contacts, development lifecycle status and performance data
 - Control access to centralSite and to the metadata for individual APIs listed in the registry
 - Model lifecycle process with each API and specify the events that are to be triggered when a lifecycle state transition happens

API Gateway

- API Gateway Service acts as a Proxy
- Consumers interact ONLY with Virtual Service
- Virtual Service
 - Created in API Gateway from 'native' service def.
 - Configured for runtime aspects using Runtime Policies
 - Activated
- Technical implementation
 - Native service can use any implementation technology
 - Virtual service relies on WSDL/SOAP or REST/XML/JSON



Software AG Training | 1 - 23

Notes:

API Gateway Capabilities

- API Definition/Authoring
- API Runtime policies
 - Threat Protection
 - Transport
 - Security
 - Transformation
 - Routing/Mediation
 - Policy Enforcement(Monitoring, Throttling, Logging)
- Consumer Applications
 - Global OAuth2, API Key, conventional identifiers support in Application
- Analytics
 - API specific & Gateway wide analytics
- API Portal integration
 - Packages & Plans(API Monetization)

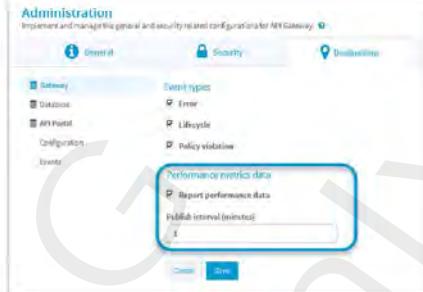


Software AG Training | 1 - 24

Notes:

Performance Metrics – Destination Administration

- Controlled from API Gateway UI
 - ‘Report Performance Data’ checkbox
 - ‘Publish Interval’ value
- Collected with every Service invocation
- Definition for each destination
 - Gateway (default setting)
 - Database
 - API-Portal
- Reported at regular intervals to selected/configured destination
- Metrics are aggregated over all consumers per configured reporting interval
 - Min/Max/Avg Response Times
 - Availability
 - Success/Error counts



The last communication scenario covered here is VS performance metrics.

Metrics include invocation counts; successful, failed and total. Also minimum, maximum and average response times.

This information is collected and published only if ‘Report Performance Data’ checkbox is checked.

Publication is for all VS or none, cannot report data for only specific VS.

Length of reporting interval is configurable, in minutes.

Metrics are collected and stored by CS, can be displayed in the CS UI, as shown on the next slide.

If Mediator attempts to send a performance data packet to CS and it fails, because CS is down, or a configuration problem, that performance data packet is lost. There is no recovery or retry.

Event Configuration – Gateway Administration

- Controlled via API Gateway
- Events are per invocation
- Events Types which are defined in Policies:
 - Transaction Events (example: logging of transactions)
- Event Types can be turned on:
 - Policy Violation (example: Violation of a security policy)
 - Error Events (example: backend service unavailable)
 - Monitoring Events (example: SLA monitoring)
- Possible destinations for Events
 - Gateway (default setting)
 - Database
 - API-Portal
- Reported per Event to selected/configured destination



Software AG Training | 1 - 26

Notes:

The screenshot shows the 'Administration' section of the webMethods Platform. The left panel, titled 'Keystores/Truststores', lists configurations for 'Ports', 'SAML', 'OAuth', 'Custom assertions', 'OAuth 2.0', 'Kerberos', 'JWT', and 'OpenID provider'. It includes sections for 'Configure keystores in API Gateway' (with entries for 'APIGatewayKeystore' and 'APIGatewayTruststore') and 'Configure truststores in API Gateway' (with entries for 'DEFAULT_TS_TRUSTSTORE', 'DEFAULT_TS_MTLS_TS_TRUSTSTORE', and 'APIGatewayTruststore'). The right panel, titled 'Ports', lists 'Configure listener ports in API Gateway' with entries for port 5005 (DefaultPrimary, HTTP, Regular) and port 5003 (APIGatewayHTTPS, HTTPS, Regular). Both panels have 'Add [item]' buttons.

- Keystore and Truststore
- Ports
- SAML, OAuth, Kerberos, JWT, OpenID configurations

API Management

WEBMETHODS API Gateway APIs Policies Applications API Packages **Administrator**

Manage APIs Create and manage your APIs. [?](#)

Add filter Name Version

Type REST SOAP

SearchCruise View API details, basic and technical information, resources and methods available, and API specifications. [?](#)

Actions [Edit](#) [Activate](#)

API details **Policies** **Applications** **Analytics**

Basic information

- Name: SearchCruise
- Version: 1.0
- Created: 2017-02-10 15:02:37 GMT
- Last updated: 2017-02-14 12:51:36 GMT
- Description: This service enables users to search for tours offered by SAGTours.

Technical information

- Native: <http://localhost:55307/SAGTours>
- endpoint(s)

Software AG Training | 1 - 28

Notes:

The screenshot shows the 'Policy Management' section of the webMethods API Gateway. The top navigation bar includes 'APIs', 'Policies', 'Applications', and 'API Packages'. The 'Policies' tab is selected. A sub-menu under 'Policies' lists 'Threat protection', 'Global policies', and 'Policy templates', with 'Threat protection' being the active tab. On the left, a sidebar lists policy categories: Global denial of service, Denial of service by IP, Denied IPs, Rules, Mobile devices and apps, and Alert settings. The main content area displays configuration for 'Threat protection rules'. It includes fields for 'Enable' (checkbox), 'Maximum requests' (text input), 'In (seconds)' (text input), 'Maximum requests in progress' (text input), 'Block intervals (minutes)' (text input), 'Error message' (text input), and 'Trusted IP addresses' (list box with an 'Add' button). A note at the top states: 'Threat protection rules are only imposed on the external port requests. No external ports are configured.' The bottom right corner of the interface shows 'Software AG Training | 1 - 29'.

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



2

API Gateway Overview

Notes:

Objectives

At the end of this chapter you ...

- Know about API Gateway Capabilities
- Can configure Users, Groups, Access Profiles in API Gateway

Notes:

API Gateway Capabilities

- API Definition/Authoring
- API Runtime policies
 - Threat Protection
 - Transport
 - Security
 - Transformation
 - Routing/Mediation
 - Traffic Monitoring (SLA, Throttle, Logging)
- Consumer Applications
 - Global OAuth2, API Key, conventional identifiers support in Application
- Analytics
 - API specific & Gateway wide analytics
- API Portal integration
 - Packages & Plans (API Monetization)



Software AG Training | 2 - 3

Notes:

Chapter Contents

- APIs
- Policies
- Applications
- Analytics
- API Packages & Plans
- Administration
- User Management

Notes:



APIs

Notes:

The screenshot shows the 'Manage APIs' page in the webMethods API Gateway. At the top, there's a navigation bar with 'WEBMETHODS API Gateway' and tabs for 'APIs', 'Policies', 'Applications', and 'API Packages'. On the right, there's an 'Administrator' dropdown and a 'Create API' button. Below the navigation, a search bar says 'SearchCruise' and 'View API details, basic and technical information, resources and methods available, and API specifications.' There are filters for 'Type' (REST and SOAP selected), 'Name' (AirportInfo), and 'Version' (1.0). To the right, there are icons for 'Edit', 'Activate', and 'Delete'. Below the search bar, there are tabs: 'API details' (selected), 'Policies', 'Applications' (highlighted with a blue box), and 'Analytics'. The main content area has three sections: 'Basic information' (SearchCruise, Version 1.0, Created 2017-02-10 15:02:37 GMT, Last updated 2017-02-14 12:51:36 GMT, Description: This service enables users to search for tours offered by SAGTours.), 'Technical information' (Native endpoint(s): http://localhost:53307/SAGTours), and 'API specific dashboards'.

Notes:

The screenshot shows the Software AG API Specific Policies interface. On the left, there's a sidebar titled "Policy catalog" with categories like Threat protection, Transport, Identify & Access, Request Processing, Routing, Traffic Monitoring, Response Processing, and Error Handling. A specific policy template, "Identity & Authenticate Application", is selected and highlighted with a blue border. The main workspace shows a flowchart of policy components: "Identity & Authenticate Application" (selected), "Transport", "Identity & Access", and "Response Processing". Arrows indicate dependencies between these components. On the right, there's a panel titled "Policy properties" for the selected template, showing conditions for "Identity & Authenticate Application" such as "Allow anonymous", "Identification Type" (HTTP Basic Authentication checked), and "Application Lookup condition" (Registered applications dropdown). Buttons for "Cancel", "Save", and "Open in full-screen" are at the top right.

- Policy Catalog – a list of categories including a number of policy templates
- A number of policies can be selected and configured for the API
 - There are policy constraints concerning dependency, possibility to have multiple policies within one category, Action Occurrence

Software AG Training | 2 - 7

Notes:



Policies

Notes:

Policy Management

- Threat Protection Policies
- Global Policies
- Policy Templates

Filter definitions for API protection from malicious attacks

The screenshot shows the 'Threat protection' configuration page under the 'Policies' section. It includes fields for 'Enable' status, 'Maximum requests*', 'In (seconds)*', 'Maximum requests in progress*', 'Block intervals (minutes)*', and 'Error message*'. There is also a 'Trusted IP addresses' section with a '+ Add' button. A note at the top states: 'Threat protection rules are only imposed on the external port requests. No external ports are configured.'

WEBMETHODS
API Gateway

Policies

Implement and manage global policies

Threat protection

Global policies

Policy templates

Global denial of service

Dental of service by IP

Denied IPs

Rules

Mobile devices and apps

Alert settings

Enable

Maximum requests*

In (seconds)*

Maximum requests in progress*

Block intervals (minutes)*

Error message*

Trusted IP addresses

+ Add

Cancel

Save

Software AG Training | 2 - 9

Notes:

Global Policies

Transaction logging
Update the basic information, associated policies, and configuration properties for the global policy. [?](#)

Policy details **Policy configuration**

Basic Information

Filters
API Type
 REST SOAP ODATA

Filter using HTTP methods (Applicable for REST only)
 GET POST PUT DELETE PATCH

Filter using API attributes
 Logical Operator
 AND OR

Attribute* API Name Operator* Equals Value* + Add

[Continue to policy configuration >](#)

- A filter can define the group of APIs for which the policy will be effective
- Policy configuration
 - Policy catalog – a subset of available policy templates
- Global Policies must be activated to be attached to APIs
 - The Policies section in an API shows global AND API specific policies

Software AG Training | 2 - 10

Notes:



Applications

Notes:

The screenshot shows the 'Applications' section of the webMethods API Gateway. A sidebar on the left lists 'WEBMETHODS API Gateway', 'APIs', 'Policies', 'Applications', and 'Packages'. The main area is titled 'MyFirstApplication' with the sub-instruction 'Update an application by providing required information.' Below this, there are tabs for 'Application details', 'Basic information', 'Identifiers', 'APIs', and 'OAuth2 Credentials'. The 'Identifiers' tab is selected and highlighted with a blue border. It contains fields for 'IP address range' (with an 'Add' button), 'Partner identifier' (with an 'Add' button), 'Client certificates' (with 'Browse', 'Delete', and 'Add' buttons), 'Claims' (with an 'Add claims set' button), and 'Other identifiers' (with a dropdown menu set to 'Hostname' and an 'Add' button). At the bottom of the application configuration, there is a table with two rows: 'Name' (Username) and 'Value' (Rebecca), and another row for 'Administrator'.

Software AG Training | 2 - 12

Notes:

The screenshot shows the Software AG Application Management interface. At the top, there's a header with the Software AG logo and the word "Application". Below the header, there's a list of tasks:

- Ensure that the right users are using the right services
 - Global Applications
 - Registered Applications (APIs are attached to the application)

A blue arrow points from the second bullet point down to a screenshot of the "Sign-up API" page. This page has tabs for "API details", "Scopes", and "Policies". It shows a table of "Selected applications" with one entry: "MyFirstApplication" (Name), "Global consumer application based on user identifiers" (Description), and "1.0" (Version). There are "Cancel" and "Save" buttons at the bottom.

At the bottom right of the interface, it says "Software AG Training | 2 - 13".

Notes:

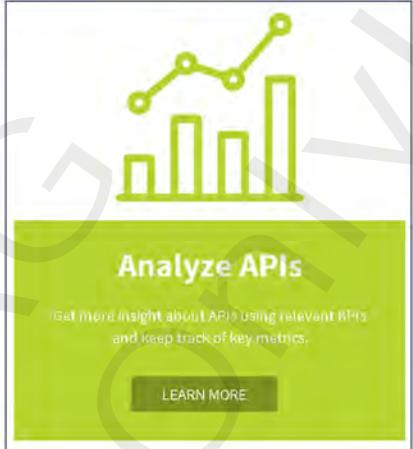


Analytics

Notes:

Analytics in API Gateway

- Analytics give Information about
 - API Gateway events
 - API-specific events
 - API trends (which APIs are more popular)
- Widgets are grouped contextually together and placed as dashboards
- Pictorial and Textual widgets are available to
 - Easily Understand
 - Analyze and compare
 - Act based on the generated data
- API Gateway support two different types of analytics
 - API Analytics -> API Gateway Provider
 - Gateway Analytics -> API Gateway Administrator



Get more insight about APIs using relevant filters and keep track of key metrics.

[LEARN MORE](#)

Software AG Training | 2 - 15

Notes:

API Analytics

- Monitor the usage of APIs
 - Performance Metrics
 - Events (Transactional, Policy Violation, Error, ...)

The screenshot shows the 'Analytics' tab of the webMethods Platform interface. On the left, a bar chart displays event counts for different categories (PerformanceData, PolicyViolation, Transaction) over the last 12 hours. On the right, a donut chart provides a summary of these categories.

Event Type	Count
PerformanceData	~45
PolicyViolation	~10
Transaction	~5

Software AG Training | 2 - 16

Notes:

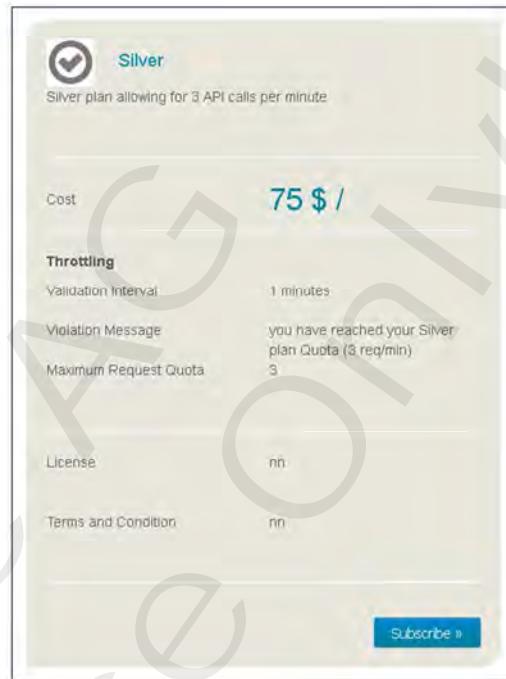


API Packages & Plans

Notes:

Monetization in API Gateway

- Exposure of tiered access to APIs with different service levels and pricing plans
- Monetization of APIs is based on Packages and associated Plans
 - A Plan defines offerings with availability guarantees, SLAs and cost structures
 - A Package is a group of APIs associated with multiple plans
- Packages and Plans can be published to API Portal for Consumers
- Consumers subscribe to a Package-Plan combination as the contract between consumer and provider



Notes:

The screenshot displays the webMethods API Portal interface. On the left, there's a sidebar with navigation links: 'Basic Information', 'Plans', and 'APIs'. Under 'Plans', three items are listed: 'Silver' (selected), 'Bronze', and 'Gold'. Under 'APIs', four items are listed: 'Human Resources', 'Production', 'Purchasing', and 'Sales Order'. The main content area shows a 'Plans' page with a 'Plans' header. It includes sections for 'Basic Information', 'Plan Details', and 'Plan Settings'. A large button at the bottom right says 'Activate'. To the right of this, there's another window titled 'APIs' which lists 'Basic Information', 'API Details', and 'API Settings' for the 'Production' API.

- Subscribing to a Plan in API Portal creates a package level access token
 - Which is applicable to all APIs associated with the package

Software AG Training | 2 - 19

Notes:

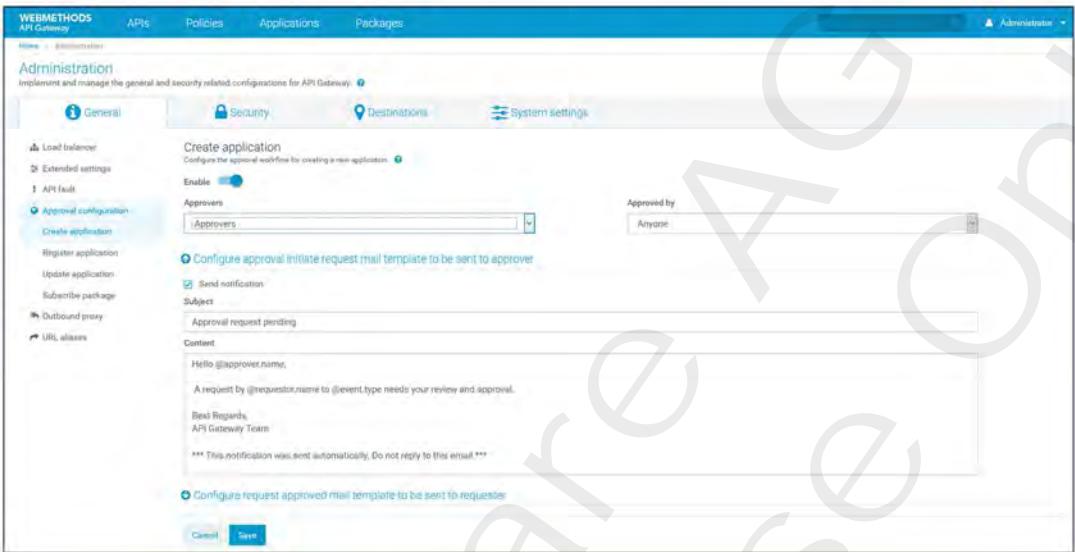


Administration

Notes:

General Administration

- Overall configuration
 - Load balancer, extended settings, proxy server alias, URL alias
 - Approval configuration on Application workflow and Package subscription



The screenshot shows the 'General' tab selected in the left sidebar under 'Approval configuration'. The main area displays settings for creating applications, including fields for 'Approvers' (dropdown menu) and 'Approved by' (dropdown menu). It also includes sections for configuring approval initiation and request approval mail templates, both with checkboxes for 'Send notification' and 'Subject' fields. Buttons for 'Cancel' and 'Save' are at the bottom.

Software AG Training | 2 - 21

Notes:

Security

- API Gateway enforces access tokens and security policies for run-time requests between application and native services
 - Corresponding security configurations need to be configured

The screenshot shows the 'Keystores' section under the 'Administration' tab. It lists two entries: 'DEFAULT_IS_KEYSTORE' (JKS type, description: 'Default keystore alias for Integration Server') and 'APIGateway/keystore' (JKS type, description: 'APIGateway Keystore'). A blue button '+ Add keystore' is visible. Below this is the 'Truststores' section, listing 'DEFAULT_IS_TRUSTSTORE' (JKS type, description: 'Default truststore alias for Integration Server') and 'APIGatewayTrustStore' (JKS type, description: 'APIGateway TrustStore'). A blue button '+ Add truststore' is also present. At the bottom, there's a section titled 'Configure keystore and truststore settings' with three dropdown menus: 'Keystore alias' (set to 'APIGatewayKeystore'), 'Key alias (signing)' (set to 'policygateway'), and 'Truststore alias' (set to 'APIGatewayTruststore').

Software AG Training

2 - 22

Notes:

Software AG Training

Security Policy Definitions

- Transport Level Authentication
 - Kerberos, HTTP Basic, OpenID, JWT
- Message Level Authentication
 - Encryption, Signature, SAML, token assertions

The screenshot shows a policy definition diagram on the left and two configuration panels on the right.

Policy Definition Diagram:

```

graph TD
    Start(( )) --> InboundAuthTransport[Inbound Authentication - Transport]
    InboundAuthTransport --> ErrorHandling[Error Handling]
    ErrorHandling --> ResponseProcessing[Response Processing]
    ResponseProcessing --> IdentityAccess[Identity & Access]
    IdentityAccess --> RequestProcessing[Request Processing]
    RequestProcessing --> LMTrafficMonitoring[LM Traffic Monitoring]
    LMTrafficMonitoring --> Routing[Routing]
    Routing --> End(( ))
    
```

Inbound Authentication - Transport Properties:

- Kerberos Tokens Authentication
- HTTP Basic Authentication
- OpenID Authentication
- JWT Authentication

Inbound Authentication - Message Properties:

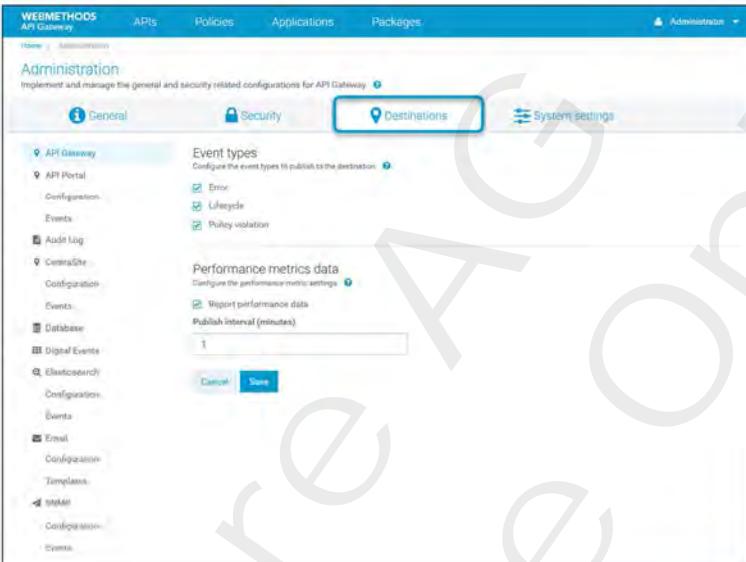
- Binding Assertion:**
 - Enter search terms to see suggestions
 - Require Encryption
 - + Add require encryption
 - Require Signature
 - + Add require signature
 - Validate SAML Audience URI
 - + Add validate saml audience uri
- Token Assertions:**
 - Require X.509 Certificate
 - Require WSS Username token
 - Kerberos Token Authentication
 - Require SAML Token
- Custom Token Assertion:**
 - Enter search terms to see suggestions
 - Requires Timestamp

Software AG Training | 2 - 23

Notes:

Destinations

- Events and performance metrics data can be published to different destinations
 - API Gateway
 - API Portal
 - Audit Log
 - CentraSite
 - Database
 - Digital Events
 - Elastic Search
 - Email
 - SNMP



Software AG Training | 2 - 24

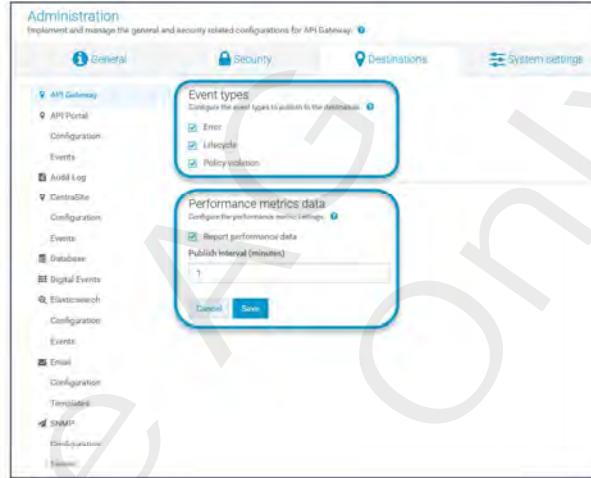
Notes:

Event Types and Performance Metrics

- Information is controlled from API Gateway UI for each destination

- Event types

- turned on/off for each destination
- additional events are defined in Policies (Transaction)
 - Destination is defined as part of the policy



- Performance metrics data

- Metrics are aggregated over all consumers per configured reporting interval

Notes:



User Management

Notes:

Welcome to **webMethods API Gateway**

Help you leverage your internal assets by exposing them as simple APIs in a secure fashion, provides relief from threat protection and complex routing and traffic management problems, provides extensive mapping and transformation support, monitors performance metrics, and provides API analytics.

Administration
Implement and manage all configurations required in webMethods API Gateway
[LEARN MORE](#)

Manage policies
Define and manage policies that perform security related actions, such as logging, audit, and performance reporting functions.
[LEARN MORE](#)

Analyze APIs
Get real-time insights about APIs using various KPIs and keep tabs of log metrics.
[LEARN MORE](#)

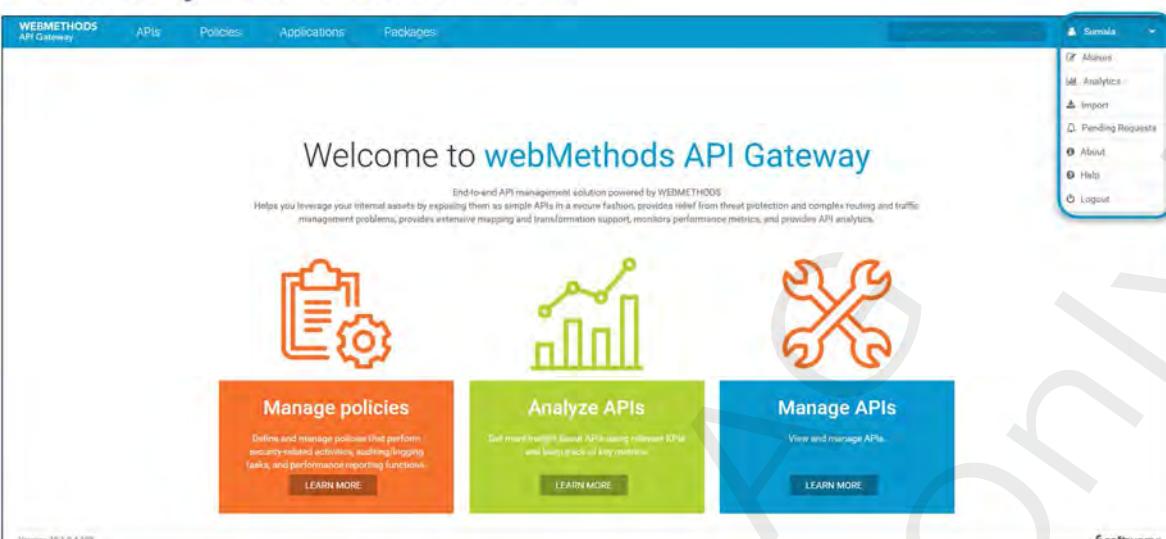
Administrator

- Administration
- Aliases
- Analytics
- Import
- Pending Requests
- User Management
- About
- Help
- Logout

Version: 12.1.0.4.105

Software AG Training | 2 - 27

Notes:

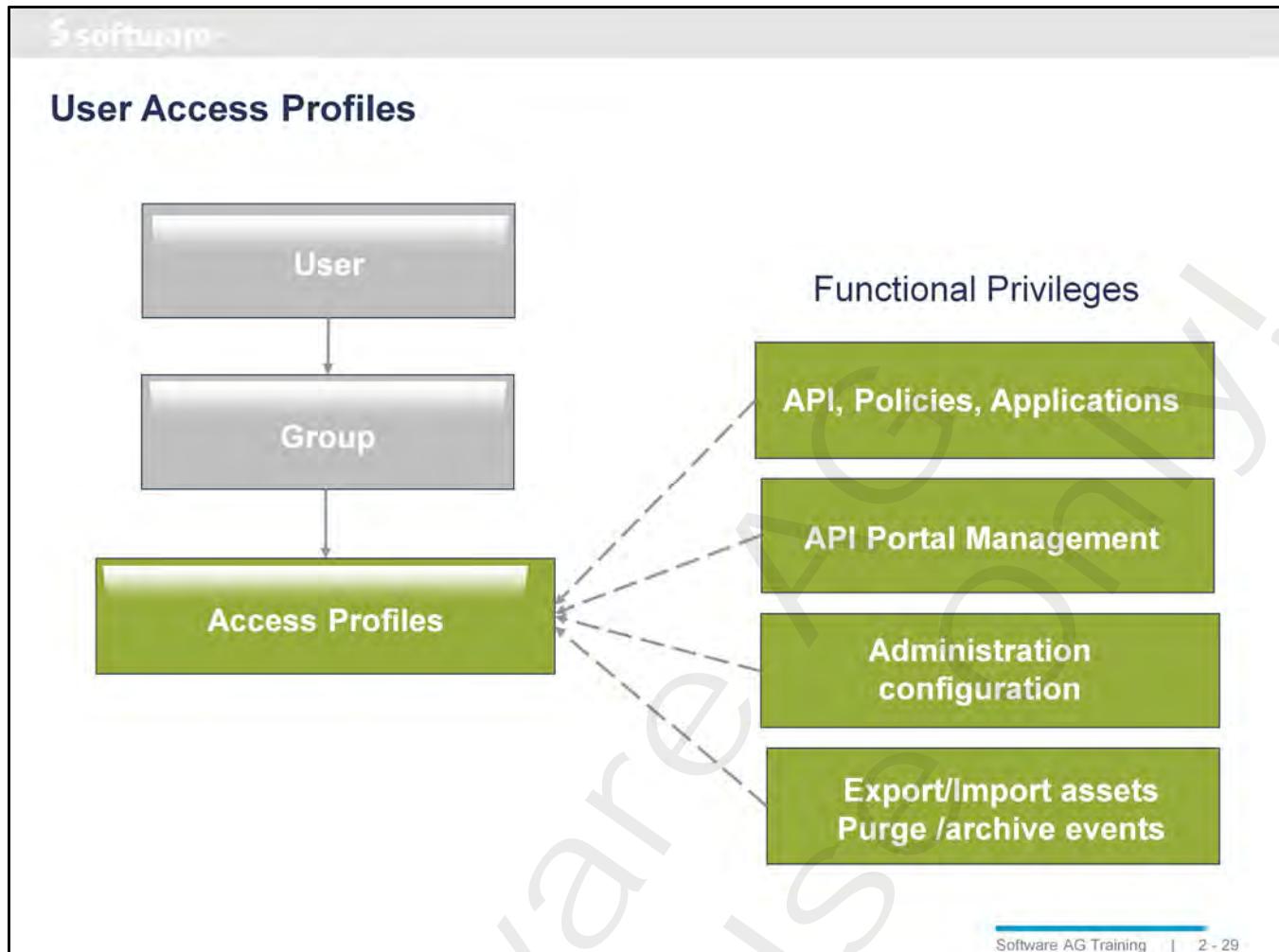


The screenshot shows the webMethods API Gateway - API Provider View. At the top, there's a navigation bar with tabs: WEBMETHODS API Gateway, APIs, Policies, Applications, and Packages. On the right side, a user menu is open, showing options like Aliases, Analytics, Import, Pending Requests, About, Help, and Logout. The main content area has a title "Welcome to webMethods API Gateway" and a subtitle "End-to-end API management solution powered by WEBMETHODS". It describes the platform as helping to leverage internal assets by exposing them as simple APIs in a secure fashion, providing relief from threat protection and complex routing and traffic management problems, offering extensive mapping and transformation support, monitoring performance metrics, and providing API analytics. Below this, there are three main sections: "Manage policies" (orange background, gear icon), "Analyze APIs" (green background, chart icon), and "Manage APIs" (blue background, wrenches icon). Each section has a brief description and a "LEARN MORE" button.

- API Gateway – global header menu
- API Provider Menu – role specific
 - Create API
 - Manage APIs
 - Analyze APIs
- User Menu (Aliases, Analytics, Import, Pending Requests, ...)

Software AG Training | 2 - 28

Notes:



Software AG Training | 2 - 29

- CentraSite provides 3 Levels of Permissions for access control on data:

Instance Level Permissions - assigned on instance-level to users & groups

Organization Level Permissions - assignable to roles

System Level Permissions - assignable to roles

- UI permissions are available for access control on UI actions (role-based UI)
- Implications between permission simplify the permission model
- Not considered here: Profile-level permissions

Users

- Definition of user information containing login ID, password and group membership
- Configuration of an external directory is supported

The screenshot shows the 'User Management' section of the webMethods API Gateway. On the left, there's a sidebar with 'User Management' and two tabs: 'Users' (selected) and 'Groups'. Below the sidebar is a list of users with their login IDs: Administrator, Andy, Peter, and Sumala. The main area is titled 'Sumala' and contains a form for updating user details. The 'Groups' tab in the main form is highlighted with a blue border. The form fields include: Login ID (Sumala), First name (Sumala), Last name (Sumus), Password (*****), Confirm password (*****), Email addresses (sumala@complex.com), Active (checkbox checked), Allow digest authentication (checkbox unchecked), and a 'Continue to associate Groups' link. There are 'Cancel' and 'Save' buttons at the top right of the form.

Notes:

Groups and Access Profiles

- A user must be associated at least with 1 access profile to login to API Gateway
- A user can only be associated with an access profile via a group

The screenshot displays two overlapping windows from the Software AG Training interface:

- Top Window (Access profile details):**
 - Groups:** This tab is highlighted with a blue border.
 - Description:** Groups associated to this access profile are allowed to access an asset based on the functional privileges assigned to this access profile.
 - Functional privileges:**
 - APIs, Policies, and Applications:** Manage APIs, Manage policies, Manage policy templates.
 - API Portal Management:** Manage packages and plans.
 - Administration configurations:** View administration configurations, Manage service result cache APIs, Manage user administration.
 - Export or Import assets and Purge and Archive events:**
 - Import assets:** Groups
 - Export assets:** API Gateway Providers
- Bottom Window (Group details):**
 - Basic information:**
 - Name:** API-Gateway-Provider
 - Description:** Users added to this group can perform specific API Gateway Provider tasks.
 - Users:** This tab is highlighted with a blue border.
 - Login ID:**
 - Email address:**

Software AG Training | 2 - 31

Notes:

API Gateway Tasks – Summary

- Design-Time
 - Expose APIs
 - Define policies and consumer applications

- Run-Time
 - Secures APIs
 - Enforces policies
 - Threat Protection, Identification, mediation, ...
 - Forwards request to native service
 - Receives response back and mediates it again
 - Forwards response to original client

Design Time

Run Time

Software AG Training | 2 - 32

Notes:



Exercise 1

- How to set up API Gateway

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



3

API Creation in API Gateway

Notes:

Objectives

At the end of this chapter you ...

- Understand the concept of API Policies
- Know how to create APIs in API Gateway
 - REST and SOAP
- Know how to activate and execute an API in API Gateway

Notes:

Chapter Contents

- Creating APIs in API Gateway
- REST Data Model
- API First – API Mocking
- SOAP APIs / OData / webSockets
- API Maintenance
- Exposing the API to Consumers
- Fine Granular Exposure of APIs to Consumers

Notes:



Creating APIs in API Gateway

Notes:

APIs in API Gateway: The Bridge for specific Consumers

The diagram illustrates the architecture of API management. It shows a Native Service (represented by a database icon with a gear) connected to an API in the API Gateway (represented by a tree-like structure). The API in the API Gateway is then connected to one or more Consumer Applications (represented by computer icons). A blue speech bubble from Andy, SOA Architect, states: "We need to avoid changes in productive services and only make them usable for specific consumers."

Native Service

API in API Gateway

Consumer Application

Andy, SOA Architect

"We need to avoid changes in productive services and only make them usable for specific consumers."

- Transform request/response
- Security
- Routing / Load Balancing
- Bridging protocols
- Performance Monitoring

Software AG Training | 3 - 5

Notes:

API Creation

- Supported API types
 - REST
 - SOAP
- API Specification Formats
 - SOAP
 - WSDL (Web Services Description Language)
 - REST
 - Swagger – Version 2.0
 - RAML (RESTful API Modeling Language) – Version 0.8
 - OData
 - webSockets
- API Creation Modes
 - Import from URL
 - Import from File
 - Single File
 - ZIP File
 - Create from Scratch

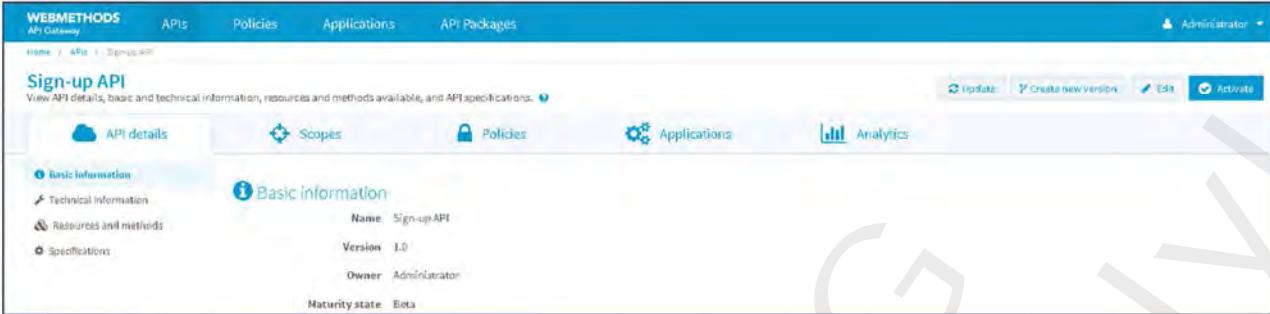
Software AG Training | 3 - 6

Notes:

The screenshot shows the 'Welcome to webMethods API Gateway' page. At the top, there's a navigation bar with tabs: 'WEBMETHODS', 'APIs', 'Policies', 'Applications', and 'API Packages'. Below the navigation, a large central area is titled 'Manage APIs' with the sub-instruction 'Create and manage your APIs.' It features a sidebar with 'ADD Filter' and 'Activation status' options. The main content area is titled 'Create API' with the sub-instruction 'Create an API by importing from a file, URL, or start from scratch.' It includes three options: 'Import API from file' (selected), 'Import API from URL', and 'Create REST API from scratch'. The 'Import API from file' section has fields for 'Select file' (set to 'petStore.json'), 'Name' (set to 'Petstore'), 'Version' (set to '1.0'), and a 'Type' dropdown set to 'Swagger'. A 'Create' button is at the bottom of this section. To the right, there's a preview pane with the title 'Description' containing the text 'This is a sample Petstore service.' At the bottom right of the main area, there's a 'Create API' button.

Software AG Training | 3 - 7

Notes:



The screenshot shows the 'Sign-up API' configuration page in the webMethods API Gateway. The top navigation bar includes links for WEBMETHODS, APIs, Policies, Applications, and API Packages, along with an Administrator dropdown. Below the navigation is a breadcrumb trail: Home > APIs > Sign-up API. A toolbar with buttons for Update, Create new version, Edit, and Activate is visible. The main content area displays the 'Basic information' section for the 'Sign-up API'. It shows the Name as 'Sign-up API', Version as '1.0', Owner as 'Administrator', and Maturity state as 'Beta'. On the left, there are tabs for Basic Information, Technical Information, Resource and methods, and Specifications. The 'Basic Information' tab is selected.

- API Management Options
 - API Details
 - Scopes
 - Policies – later in this class
 - Applications – later in this class
 - Analytics – later in this class
- API Lifecycle States
 - Edit Mode
 - Read-Only Mode
 - Active Mode

Software AG Training | 3 - 8

Notes:



Exercise 2

- Creating a REST API using a Swagger file

Notes:



REST Data Model

Notes:

REST (Representational State Transfer)

- Architectural style centered around transferring representations of resources
- REST is ...
 - ... more data-centric than SOAP
 - ... gaining acceptance as a light-weight alternative to SOAP
- There are very few standards around REST, so much of its use is driven by conventions
- URL points to a resource, not a service: *http://xxx/customer*
- HTTP method defines what to do with the resource
 - GET/PUT/POST/DELETE
- URL path often contains ID information: *http://xxx/customer/123*

Notes:

REST API Terminology

- Resource
 - The Objects the API is dealing with eg. customers, cruises, accounts
- Resource Path
 - The resource specific path of the URL identifying the resource
 - Eg <base url>/customers
- Base URL
 - Entry URL to the API without resources
- Method
 - The HTTP verb used on a resource to perform a CRUD operation

```
GET http://xxx.com/api/v1/customers/1234
  _____
  |       |
  Verb   Base URL  Resourcepath Path
  |       |           |
  |       |           Parameter
```

Software AG Training | 3 - 12

Notes:

How to identify API candidates?

- Identify your business motivation for creating / offering APIs
 - Examples:
 - Allow API access to product catalog as new e-commerce channels
 - Offer location APIs to allow locating retail stores or agencies
 - Sell digital content through APIs
 - Foster 3rd party innovation – eg. 3rd party creation of mobile apps or mashups
 - Simple B2B integration with SMBs
- Identify your targeted audience
 - Examples:
 - Unknown 3rd party mobile developers
 - B2B trading partners

Clearly documenting business motivation for an API will help identify initial candidates

Software AG Training | 3 - 13

Notes:

API Resources & Operations

- Think of “Resources” as the Business Objects that the API has to support
- Example: customer Resource
- Possible operations on “customers” resource
 - Adding a new customer
 - Getting a list of all existing customers
 - Getting details on a specific customer
 - Updating a customer detail information
 - Deleting a customer

Notes:

Designing APIs

- Good APIs are designed in a hierarchical way
 - Eg
 - /customers
 - /customers/{cust_id}/orders
 - /customers/{cust_id}/orders/{order_nr}
- APIs are structured by their resources, NOT by operations
- HTTP methods/verbs describe the operations possible on a resource
 - Getting all customers
`GET http://xxx/customers`
 - Getting a specific customer
`GET http://xxx/customers/4711`
 - Create a new customer
`POST http://xxx/customers`
 - Update a specific customer
`PUT http://xxx/customers/4711`

Notes:



Exercise 3

- Identify Business Objects and configure REST APIs

Notes:

Example: SAGTours

- SAGTours is Travel agency specialized on organizing and selling cruises.
They operate a traditional travel agency chain as well as an e-commerce site
- SAGTours wants to establish APIs as new sales channels to support 3rd parties creating mobile apps and innovative new travel applications leveraging SAGTours offerings in the backend
- Primary usecases around cruise tour booking
 - Users need to search for available cruises and tours
 - Users need to be able to retrieve details about a selected cruise
 - Users need to be able to book a tour
 - Users might need to open an account with SAGTours for billing
- Resulting APIs
 - Search API
 - Booking API
 - SignUp API

Notes:

Creating a REST Service from Scratch

- Create REST service
 - Name
 - Version
 - Maturity state, API grouping
 - Pre-defined but configurable list of states
 - Description

Software AG Training | 3 - 18

Specifying a Native Endpoint

APIs (Service, XML service and REST service) can contain one or more operations or resources.

For an instance of a XML or REST API only. If you are using XML schemas along with your XML / REST API, attach the schema file to the catalog entry using Attach () in the API's actions menu.

CentraSite automatically populates the schema URL and the associated resources in the Technical Details profile.

After you have specified a schema, specify the following:

Attribute Description Endpoint An endpoint for the API that allows consumers of the API to find and communicate with the API. Namespace A binding namespace for the endpoint. Resource A name for the resource. You can specify multiple resources for an endpoint. HTTP Method HTTP request method(s) for bridging protocols (GET, POST, PUT, DELETE).

Namespace definition for REST services

- Mandatory for Sopa based web services
- Optional for REST services, but the REST service is defined for Mediator in a wsdl. It is good practice to use a namespace. A suggestion which how to set namespace use a combination of Organization/Service-Name. In the future for REST services this will be an optional property.

Now also available in native service:

Runtime Metrics and RunTime Events

Runtime Metrics

Displays the run-time performance metrics associated with the API. If you are using webMethods Mediator, webMethods Insight or another run-time monitoring component to log performance metrics for an API, CentraSite displays those metrics on this profile.

Runtime Events

Displays the run-time events associated with the API. If you are using webMethods Mediator, webMethods Insight or another run-time monitoring component to log run-time events for an API, CentraSite displays those events on this profile

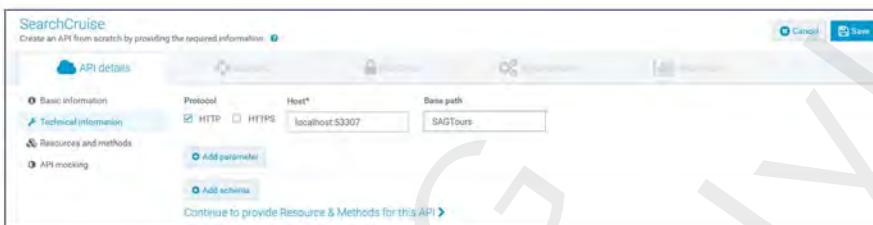
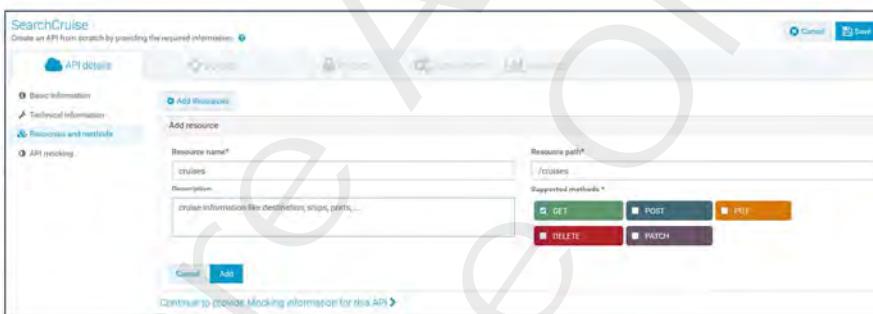
New profile Provider Overview

Displays the list of native and virtual endpoints defined for the API. In this profile, a native endpoint is represented by the *Binding*, and a virtual endpoint is represented as an *Alias* that identifies a specific Access URI (i.e., address where the virtual API is published). This profile also contains control for viewing the enforcement definition of a virtual endpoint.

Providing Technical Details and Resources / Methods

- Provide Technical Details
 - Endpoint, Sandbox
 - Namespace
 - Content Type
 - API Parameters

- Add Resources
 - Resource name and Path
 - Resource Description
 - Select Method
 - GET, POST, PUT, DELETE, PATCH

Software AG Training | 3 - 19

Specifying a Native Endpoint

APIs (Service, XML service and REST service) can contain one or more operations or resources.

For an instance of a XML or REST API only. If you are using XML schemas along with your XML / REST API, attach the schema file to the catalog entry using Attach () in the API's actions menu.

CentraSite automatically populates the schema URL and the associated resources in the Technical Details profile.

After you have specified a schema, specify the following:

Attribute Description Endpoint An endpoint for the API that allows consumers of the API to find and communicate with the API. Namespace A binding namespace for the endpoint. Resource A name for the resource. You can specify multiple resources for an endpoint. HTTP Method HTTP request method(s) for bridging protocols (GET, POST, PUT, DELETE).

Namespace definition for REST services

- Mandatory for Sopa based web services
- Optional for REST services, but the REST service is defined for Mediator in a wsdl. It is good practice to use a namespace. A suggestion which how to set namespace use a combination of Organization/Service-Name. In the future for REST services this will be an optional property.

Now also available in native service:

Runtime Metrics and RunTime Events

Runtime Metrics

Displays the run-time performance metrics associated with the API. If you are using webMethods Mediator, webMethods Insight or another run-time monitoring component to log performance metrics for an API, CentraSite displays those metrics on this profile.

Runtime Events

Displays the run-time events associated with the API. If you are using webMethods Mediator, webMethods Insight or another run-time monitoring component to log run-time events for an API, CentraSite displays those events on this profile

New profile Provider Overview

Displays the list of native and virtual endpoints defined for the API. In this profile, a native endpoint is represented by the *Binding*, and a virtual endpoint is represented as an *Alias* that identifies a specific Access URI (i.e., address where the virtual API is published). This profile also contains control for viewing the enforcement definition of a virtual endpoint.

Name*	Description	Type	Datatype	Required	Repeat	Value
start-date	Cruise start date, in the format YYYYMMDD. Selects cruises that start on this date. Combined with endDate, this parameter filters...	Query-string	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2017-01-01

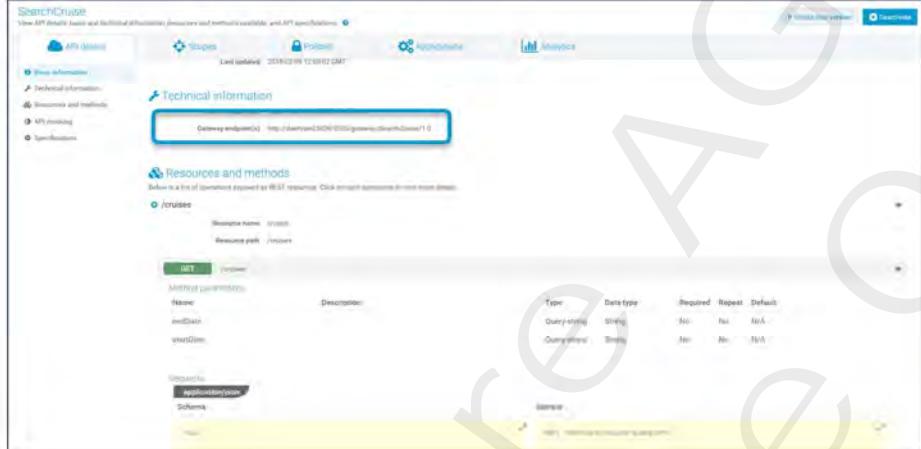
- Parameters are used to pass and add additional information to a request.
 - They can be added at the API , REST Resource & REST Method levels.
- Details:
 - Parameter Name and Description
 - Parameter Type
 - Header
 - passed as a part of custom HTTP headers , contain metadata for client or server
 - Query
 - much like attributes, specified while accessing the API
 - Path
 - integral part of the URL, not available on Method level
 - Data Type
 - Values – Required / Array / Possible / Default

Software AG Training | 3 - 20

Notes:

Activating the Service in API Gateway

- Creates a proxy service on Integration Server
 - Gateway endpoint(s)
- API Gateway Service – representation of proxy service, in which routing to the native service is defined



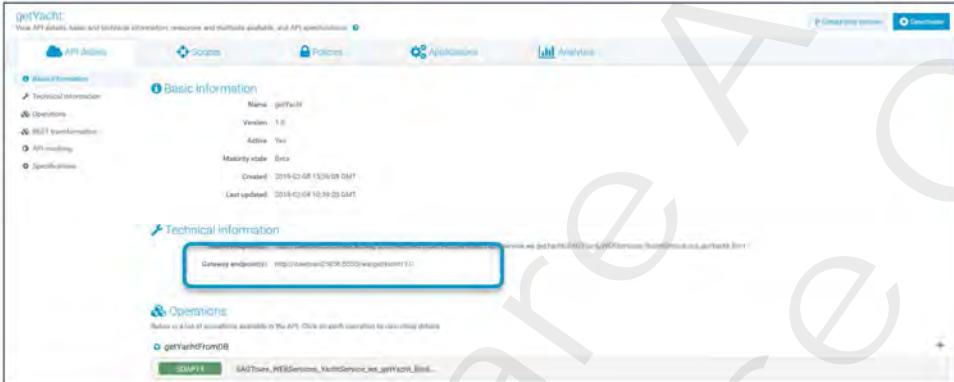
The screenshot shows the 'Technical information' tab of an API Gateway Service configuration. It displays the 'Gateway endpoint(s)' as 'http://localhost:8080/generic/direct/HelloWorld'. Below this, under 'Resources and methods', there is a single method entry for '/HelloWorld'. The method details show it is an 'HTTP' method with 'GET' selected. It has two parameters: 'method' (Query-string, String, No, No) and 'username' (Query-string, String, No, No). A note at the bottom states: 'Below is a list of operations exposed as REST resources. Click on each resource to view more details.'

Software AG Training | 3 - 21

Notes:

Activated APIs on Integration Server

- REST APIs
 - Gateway endpoint points to Integration Server gateway
 - <http://localhost:5555/gateway/Sign-upAPI>
- SOAP based APIs
 - Gateway endpoint points to Integration Server web-services stack
 - <http://localhost:5555/ws/AirportInfo>



Software AG Training | 3 - 22

Notes:

Deactivate API Gateway

- Deactivating an API
 - Deletes API on Integration Server
 - Removes Gateway endpoint definition in API details page
 - Enables configuration of Policies

The screenshot shows two views of the 'SearchCruise' API details page. The top view shows the 'Basic information' section with a 'Deactivate' button highlighted with a blue border. The bottom view shows the 'Technical information' section, which is also highlighted with a blue border. Both sections include fields like Name, Version, Owner, Active status, and API endpoint details.

Software AG Training | 3 - 23

Notes:

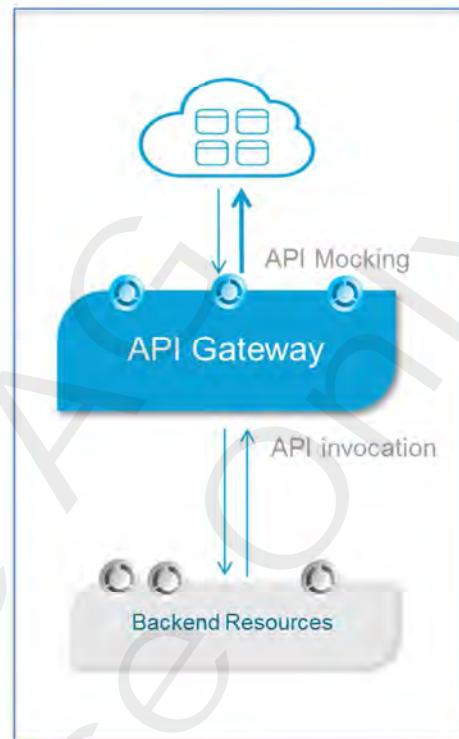


API First – API Mocking

Notes:

API Mocking

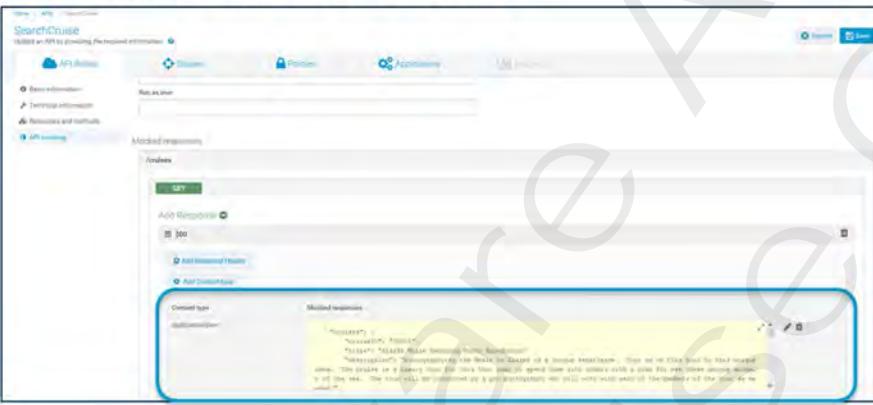
- API Gateway simulates the native API
 - By providing response messages defined in API Gateway
 - No call to the native service
- Use cases
 - API First
 - Start with the Design First approach
 - Testers and consumers can work in parallel
 - Reduce time from design to production
 - Replacement version of API



Notes:

Enabling API Mocking

- API Mocking State
 - On / Off
 - Mocking enabled: API Gateway sends a mock response
- A Mock Response can be configured for each combination of
 - resources and methods or operation definitions
 - response code
 - Content type
 - as schema or sample payload



The screenshot shows the 'Mocked responses' configuration for an API resource named 'SearchCruise'. It displays a table with columns for 'Method', 'Content type', and 'Mocked responses'. One row is selected, showing a preview of the response body: 'Placeholder(s) are used in the response body. You can click here to edit the placeholder(s). The placeholder(s) will be replaced by a pre-emptive response payload when the request is received.' Below the table, there are buttons for 'Add Response' and 'Add Header'.

Software AG Training | 3 - 26

Notes:

Dynamic Mock Response

- Mock payload is based on the request
 - Mock payload allows for variable request information
- Mock definition can have condition parameters
 - Header
 - Query
 - Request Body

Software AG Training | 3 - 27

Notes:

ESB Service Sending Mock Response

- Custom logic in Integration Server service to handle request

The screenshot shows the 'API Mocking Demo' interface. At the top, there are tabs for 'API details', 'Scopes', 'Policies', and 'Applications'. The 'API mocking' tab is selected. Under 'Basic information', the 'Invoke service' field contains 'api@gateway.specifications.sample.mock'. Under 'Run as user', it says 'Administrator'. Below this, there's a large test window. The method is set to 'PATCH' and the URL is 'http://MCSURG01:5555/gateway/API%20Mocking%20Demo/v2'. The 'Body' tab is selected, showing the response body: 'TEST ESB SERVICE : STRING TO UPPER'. The status bar at the bottom right indicates 'Status: 200 OK' and 'Time: 33 ms'. There are buttons for 'Send' and 'Save'.

Notes:

Priority of a Mock Response

- Default mock response is configured for each combination of resource, operation, status code, and content-type based on the example and schema specified in that API
- Priority of generating the default mock response at design-time:
 - Sample
 - Schema
- Priority of sending a mock response at run time
 - Condition based
 - ESB service
 - Default mock response

Notes:



Exercise 4

- Creating Rest API from Scratch

Notes:



SOAP APIs / OData / webSockets

Notes:



API Maintenance

Notes:

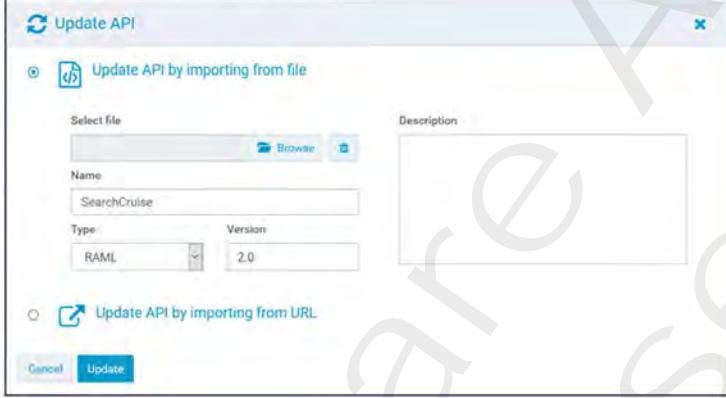
▪ Deactivate API to allow changes to the API
▪ Switch to Edit Mode for changes

▪ Change API Details
▪ Save changes, activate API

Notes:

Update API

- Update an API by importing an API definition (WSDL, Swagger or RAML)
 - From a file
 - From a URI
- Overwrites an existing API definition, retains the exposure to consumer settings
 - Existing Scope definitions
 - Configured Policies
 - REST enabled Path definitions



Software AG Training | 3 - 34

Notes:

The screenshot shows the Software AG API Management interface. On the left, there's a sidebar titled "Manage APIs" with filters for "Type" (REST, SOAP) and "Activation status". In the center, there's a list of APIs with one named "SearchCruise" selected. The "Basic information" tab is open, showing details like Name: SearchCruise, Version: 2.0, Owner: Sumala, Active: Yes, Maturity state: Production, and API grouping: Search. Below this, the "Technical information" tab is shown, containing the native endpoint (<http://localhost:55007/SABToys>) and gateway endpoint (<http://dweltzam25856:5555/gateway/SearchCruise/2.0>). At the top right, a modal window titled "Create version" is open, showing a version number of 2.0 and a checked "Retain applications" option. Buttons for "Cancel" and "Create" are at the bottom of the modal.

Notes:



Exposing the API to Consumers

Notes:

API Gateway Services

- Services in API Gateway act as proxies for native services
- Switch protocols (HTTP/S, SOAP 1.1/1.2)
- Support different routing methods
- Rules for Security, Monitoring, Logging, Validation, ...
- Transform request/response to ensure compatibility

Consumer Application Systems

Provider System Services

Serv A

Receive	Require HTTP
Enforce	Security
Routing	context-based
Response	Transformation

Serv A

Receive	Require HTTPS
Enforce	...
Routing	JMS Routing
Response	Error handling

Serv A

Receive	Require HTTP
Enforce	Monitoring
Routing	
Response	add custom h.

Software AG Training | 3 - 37

Notes:

Policies

- Visualization of Message Flow enforcing Policy Actions
- 8 different Blocks to configure the Policy Actions
- Autoconfigured for straight-through processing based on service definition
 - in Transport Definition: Protocol and Soap Version
 - In Routing Definition: native endpoint Definition

The screenshot shows the webMethods Policy catalog interface. On the left, there's a sidebar with a tree view of policy categories: Threat protection, Transport (selected), Identify & Access, Request Processing, Routing, Traffic Monitoring, Response Processing, and Error Handling. The main area displays a message flow diagram with several policy blocks connected by arrows. From top-left to bottom-right, the blocks are: Requester (HTTP), Enriching, Threat detection, Identify & Access, Response Processing, Use traffic monitoring, and Routing. A green box highlights the 'Requester (HTTP)' block. A blue box highlights the 'Routing' block. On the right side of the screen, there's a panel titled 'Policy properties' with settings like 'Require HTTP / HTTPS', 'Protocol: HTTP', and 'SOAP version: SOAP 1.1'. At the bottom right, there's a 'Software AG Training | 3 - 38' footer.

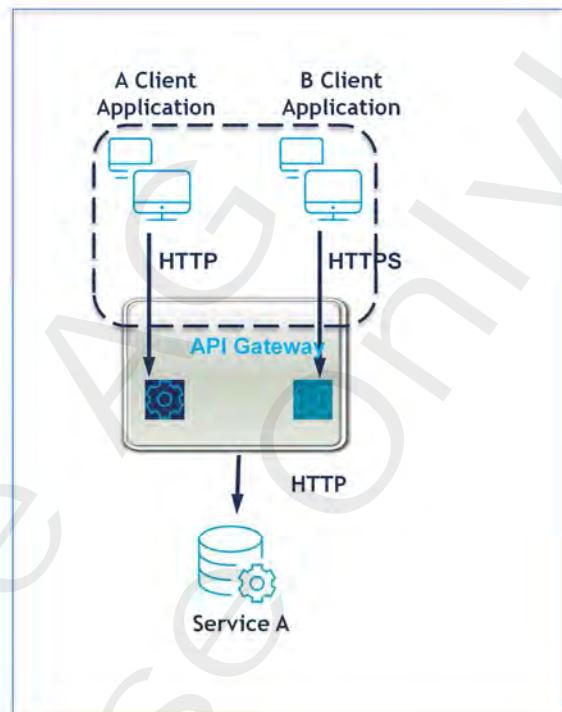
Notes:

Message Flow

- Security and Authentication (Identify & Access, Threat Protection)
 - Transport and message-layer security (authentication, authorization, ...)
 - DMZ-strength security between consumers apps and internal servers (firewall and mediate the APIs)
- Transformation and Routing (Transport, Request Processing, Routing, Error Handling, Response Processing)
 - Flexibility in message formats and transports for consumers
 - Dynamic routing of requests based on payload content, context, straight-through ...
- Runtime events (Traffic Monitoring)

Request Handling

- Transform request to ensure compatibility
 - Switch Protocols
 - HTTP(s) <-> HTTPS
 - Request Transformation
 - invoke XSL transformation file
 - Invoke webMethods IntegrationServer
 - invoke Integration server service
 - Set Media Type
 - Content-type for request if not specified in the request header



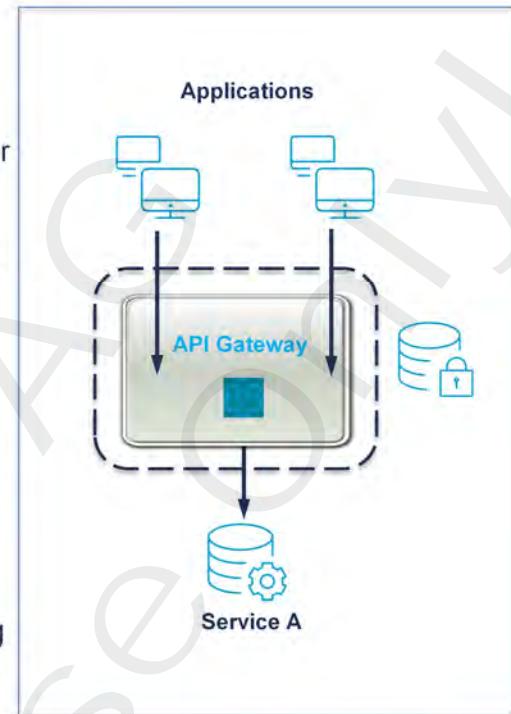
Identification Management

API Security Policies

- Client Validation
 - Authentication
 - Confirm identity via digital certificate, password or token
 - Authorization
 - Limit access to data & services based on identity
- Confidentiality
 - Encrypt data to prevent interception
- Integrity
 - Ensure data has not been tampered with en route

Application <-> API Mapping

- How to identify and validate clients who are trying to access the API as consumers in API Gateway



Software AG Training | 3 - 41

Notes:

Traffic Management

Logging & Monitoring

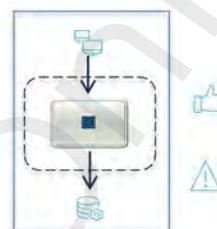
- Monitor and collect information about the number of messages processed (success or failure, number of calls, average/min/max response time)
 - Log Invocation
 - Monitor Service Level Agreement
 - Monitor Service Performance

Traffic Management

- Avoid overloading the back-end services
 - Throttling Traffic Optimization (limit the number of calls)
 - Service Result Cache

Validate Schema

- Validate all XML request and/or response message against an XML Schema
 - Validate Schema



Notes:

Routing Management

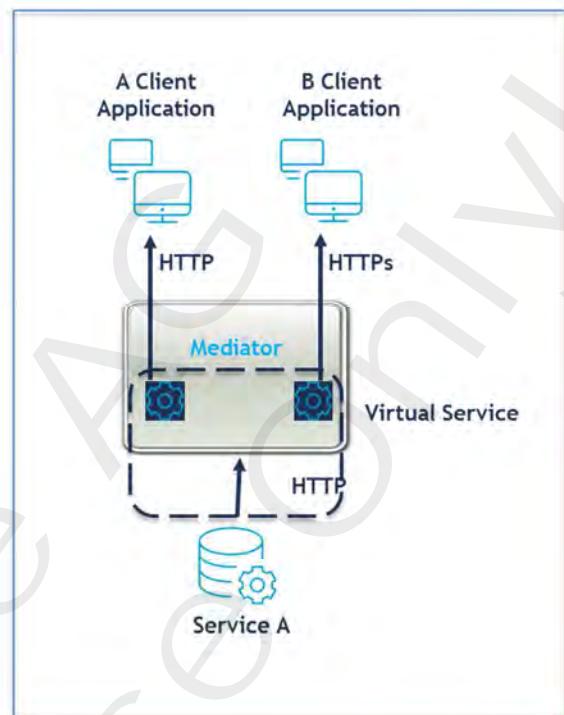
- Outbound Authentication Actions Transport / Message
 - Verify that the client has the proper credentials to access the API
 - HTTP Basic Authentication
 - Kerberos Authentication (Transport & Message level)
 - NTLM Authentication
 - OAuth2 Authentication
 - JWT
 - SAML Authentication (Message level)
 - WSS Username (Message level)
- Routing Actions
 - Route the incoming message
 - Content Based Routing
 - Context Based Routing
 - Custom HTTP Header
 - Load Balancer Routing
 - Dynamic Routing
 - JMS Routing / JMS Properties



Notes:

Response Processing and Error Handling

- Transform response to ensure compatibility
 - Invoke webMethods IntegrationServer
 - invoke Integration server service
 - Request Transformation
 - invoke XSL transformation file
 - Validate Schema
- Error Handling
 - Error Conditions
 - Failure Message
 - Custom Error Variables
 - Pre- and Post-Processing

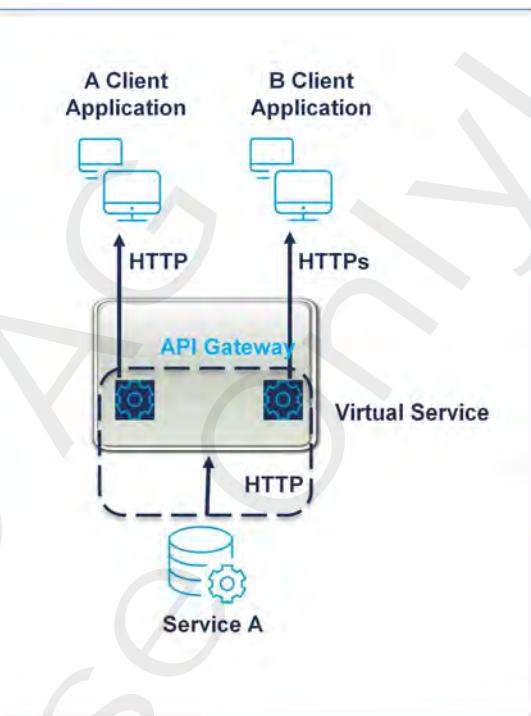


Software AG Training | 3 - 44

Notes:

Response Processing and Error Handling

- Transform response to ensure compatibility
 - Invoke webMethods IntegrationServer
 - invoke Integration server service
 - Request Transformation
 - invoke XSL transformation file
 - Validate Schema
- Error Handling
 - Error Conditions
 - Failure Message
 - Custom Error Variables
 - Pre- and Post-Processing



Software AG Training | 3 - 45

Notes:



Exercise 5

- Adding a Logging Policy

Notes:



Fine Granular Exposure of APIs to Consumers

Notes:

Requirement: Fine Grain Control of APIs

- Expose a subset of resources or operations to consumers.
 - To control the visibility of specific resources and operations
- Default behaviour when activating an API
 - All resources and methods are visible
 - All operations are visible
- Feature to activate a subset of functionality within an API
- This feature can be used to control the visibility of specific resources/operations within an API.
 - REST API: the visibility state can be set for any resource or that of methods within the resource
 - SOAP API: the visibility state can be set for each operations individually

Notes:

Text for speech :

As soon as we activate an API all its resources become accessible to the consumers of the API.

This feature is designed to give fine grained control over which parts of an API can be accessed by the consumers.

If we intend to expose only few resources within an REST API, or just a few methods within the resource of the API, we can do that using this feature. Activating the API after that will expose only the chosen resources or methods within the API.

In case of a SOAP API, we can choose the operations that needs to be exposed to consumers.

Expose to Consumers – REST API

- Activate REST API
 - By default all resources and methods to access the API are visible to consumers
- Expose to Consumers
 - Control the visibility either for an entire resource or for individual method.
 - Visibility of a resource in an active API depends on whether **Expose to consumers** is on or off



Software AG Training | 3 - 49

Notes:

Text for speech:

Every REST API has one or more resources and each of these have different methods to access them.

Activating the API will make all its resources and methods become accessible to consumers, this is because all of the resources and methods are exposed to consumer by default.

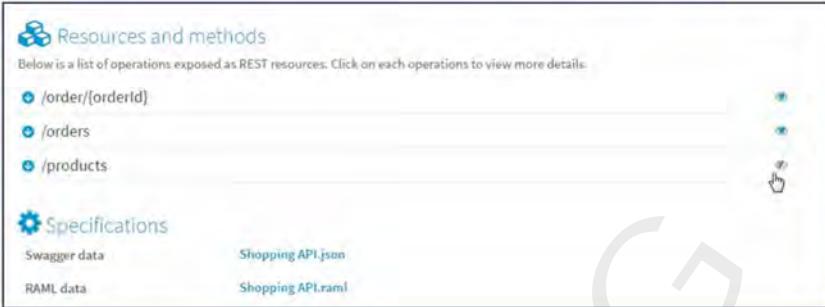
If a specific resource is not exposed to consumers, then all its methods are not exposed to consumers. However even if one method remains exposed to consumers the resource stays exposed to the consumers.

Invoking a non exposed resource or method will return a 404 error code.

The Swagger or RAML specification of the API will not show the resources that are not exposed to the consumers

If none of the API's resources are exposed to consumers, then it will fail the activation of API.

Limitation on Expose to Consumers for REST APIs

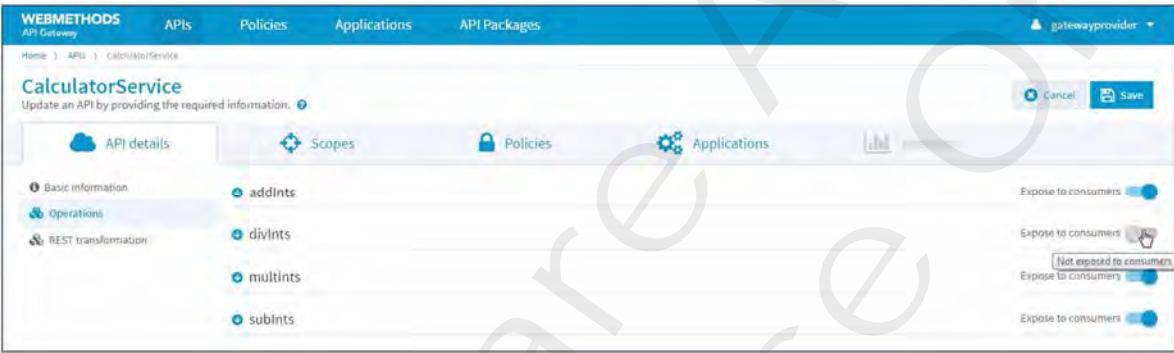


- Control the visibility either for an entire resource or for individual method
 - Any invocation of a resource or method that is not exposed to the consumer will return a 404 status
 - Resources and methods that are not exposed to consumers will not be part of the Swagger or RAML specification
- Limitations
 - A methods cannot be in 'exposed to consumer' while the resource is not exposed
 - To activate an API at least one resource should be exposed to the consumer

Expose to Consumers – SOAP API

- Activate SOAP API
 - By default all operations are visible to consumers

- Expose to Consumers
 - Control the visibility for individual operations
 - Visibility of a resource in an active API depends on whether **Expose to consumers** is on or off



The screenshot shows the 'CalculatorService' configuration page in the webMethods API Gateway. The 'Expose to consumers' switch is turned on for all four operations listed: addInts, divInts, multInts, and subInts. The 'Not exposed to consumers' option is also present but not selected.

Software AG Training | 3 - 51

Notes:

Text for speech:

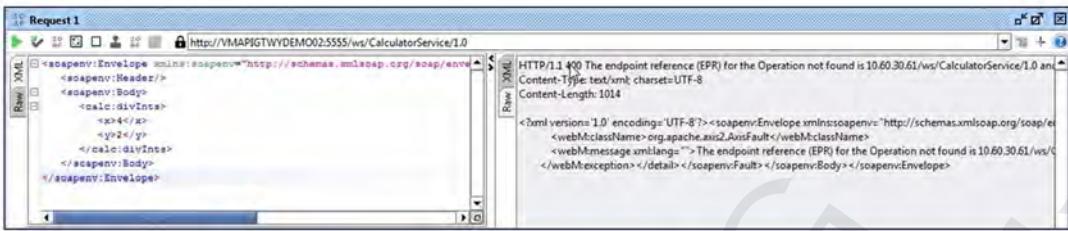
Every SOAP API has one or more operations to access the API
 Activating the API will make all its operations become accessible to consumers. This is because all of the operations are exposed to consumers by default.

It is not possible to Activate a SOAP API if none of its operations are exposed to consumers.

If an operation is not exposed to consumer, it will not be listed in the wsdl specification of the API.

Further, only those operations which are exposed to consumers will be eligible for SOAP to REST Transformation.

Limitation on Expose to Consumers for SOAP APIs



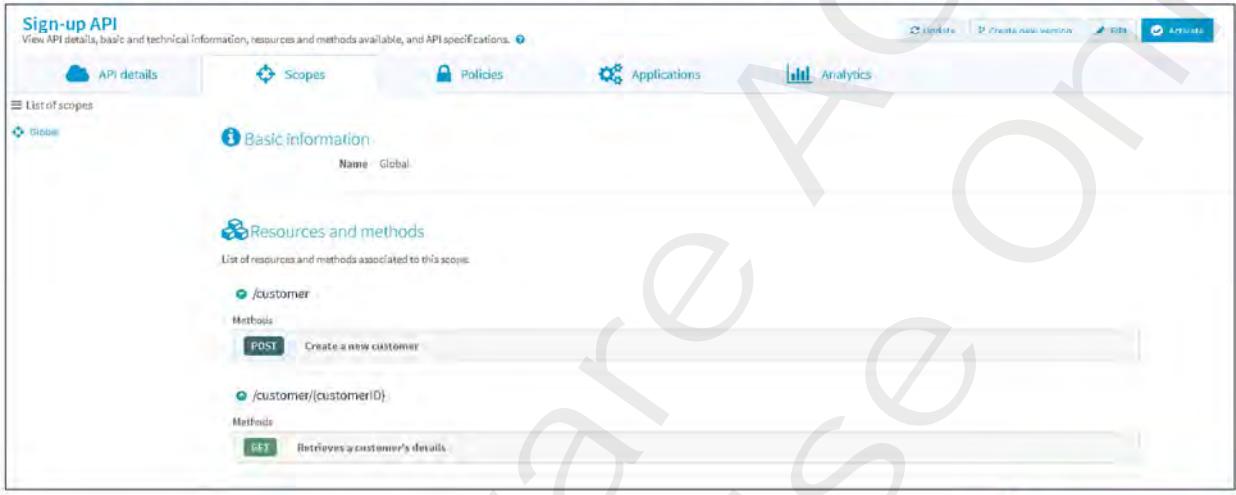
- Control the access of every operation individually.
 - Any invocation of an operation that is not exposed to the consumer will return a 400 status.
 - The wsdl specification of the API will not show the operations that are not exposed to consumers.
- Limitations
 - To activate an API, at least one operation should be exposed to consumers.
 - SOAP to REST Transformation can be done only for operations that are exposed to consumers

Software AG Training | 3 - 52

Notes:

Scopes

- A logical grouping of
 - REST resources, methods, or both
 - SOAP operations
- Reduce complexity by defining a Scope and creating scope level policies
 - Fine granular definition of policies for each individual scope



The screenshot shows the 'Sign-up API' interface. At the top, there are tabs for 'API details', 'Scopes' (which is selected), 'Policies', 'Applications', and 'Analytics'. On the left, there's a sidebar with 'List of scopes' and a single entry 'Global'. The main content area has two sections: 'Basic information' (Name: Global) and 'Resources and methods'. Under 'Resources and methods', it lists '/customer' (Method: POST, Description: Create a new customer) and '/customer/{customerId}' (Method: GET, Description: Retrieves a customer's details).

Software AG Training | 3 - 53

Notes:

Creating a Scope

- Optional property
 - An API can have 0 to n Scopes
 - A scope can have 0 to n resources, methods, or operations
- Use Case:
 - Enforce different authentication mechanisms for different REST resources and methods depending on access level
 - For example a WRITE scope could be used to enforce a higher-level of secured access and manipulation of the REST data

Notes:



4

Identification Management

Notes:



Objectives

At the end of this chapter you ...

- Understand the concepts of
 - Identification and Access Management
- Can define different security mechanisms to protect the API
 - Different Identification Types

Notes:

Chapter Contents

- Identification and Access Management
- Inbound Authentication
- Inbound Authentication Message Level
- Authorize User

Notes:



Identification and Access Management

Notes:



The Four Pillars of Security

- Authentication
 - The process of verifying the identity or other attributes of a user, user device, or other entity. This is a prerequisite of authorization.
- Authorization
 - Access privileges granted to a user, program or process.
- Confidentiality
 - Information is not disclosed to users, user devices or processes unless authorized (i.e., Encrypt data to prevent interception).
- Integrity
 - The property whereby an entity has not been modified in an unauthorized manner. Ensuring information non-repudiation and authenticity.

Notes:

Security Run-time Policies

HTTP Basic Authentication
Kerberos Token
OpenID Authentication
JWT Authentication

API Key
OAuth2 Token / Kerberos Token
Hostname Address / IP Address
Range
HTTP Basic Authentication
JWT
OpenID Connect
SSL Certificate
XPath expression
Token Assertions (SOAP)

- X.509 Certificate,
- WSS Username Token,
- Custom Token Assertion

Require Encryption (SOAP)
Require Signing (SOAP)
Require Timestamp (SOAP)

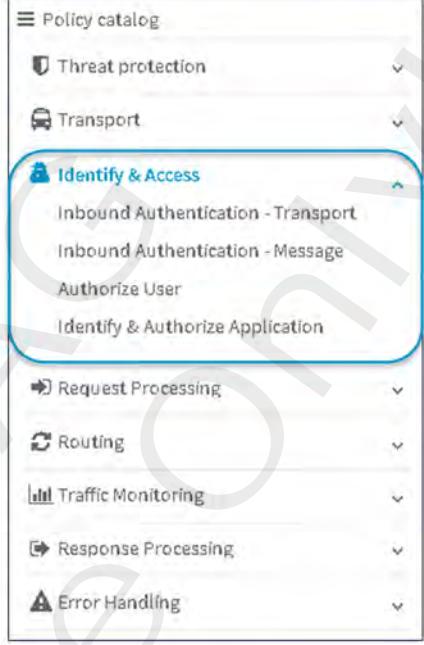
The diagram illustrates the flow of security policies through an API Gateway. It starts with three policy boxes on the left: 'Authentication By Transport' (user icon), 'Identification & Authorization' (app icon), and 'Authentication By Message' (file icon). Dashed arrows from these boxes point to a central 'API Gateway' box. From the 'API Gateway' box, solid arrows point to three cloud icons representing different protocols: REST (top), REST (middle), and SOAP (bottom).

Software AG Training | 4 - 6

Notes:

Identify and Access in API Gateway

- Different ways of identifying and authenticating the consumer accessing the API
 - Inbound Authentication – Transport
 - Inbound Authentication – Message
 - For SOAP APIs
- Different ways of authorizing the consumer accessing the API
 - Authorize User
 - Authorize consumer based on a list of Users or a list of Groups
 - Identify and Authorize Application
 - Authorize application and identify the consumer as being part of the application based on Application Identifiers



Software AG Training | 4 - 7

Notes:

Authorization

- Verify that the service consumer is authorized to invoke the service
 - Identify & Authorize policy action
 - Requires => Application Lookup Condition
 - Mapping to a global consumer application
 - Mapping to a registered consumer application of the service
 - Authorize User policy action
 - Requires => Inbound Authentication policy action
 - List of Users
 - List of Groups
- API Gateway maintains a list of consumer applications specified by a list of Identifiers
 - different Identifiers Types are available to identify and authenticate the Consumer
 - Hostname, IP Address, Username, custom XPath expression, API Key, OAuth2 token ...

Notes:



Inbound Authentication

Notes:

Transport Security

- User credentials and claims are passed by using the transport layer
 - Transport-dependency allows fewer authentication options
- Each transport protocol (TCS, IPC, HTTP, MSMQ) has its own mechanism for passing credentials and handling message protection
 - Most common approach: use Secure Sockets Layer (SSL) for encrypting and signing the contents of the packets sent over Secure HTTP (HTTPS)
- Advantage:
 - Provides interoperability
- Disadvantage:
 - Point-to-point security with no provision for multiple hops
 - Transport dependent upon the underlying platform, transport mechanism and security provider, such as NTLM or Kerberos

Notes:

Inbound Authentication - Transport

- Security based on Transport level
 - To protect the proxy service based on user authentication
 - Available for REST and SOAP APIs
 - Methods
 - HTTP Basic Authentication
 - Kerberos Authentication
 - OpenID Authentication
 - JWT (JSON Web Token) Authentication

The screenshot shows the 'getYacht' API details page. On the left, the 'Identify & Access' section is expanded, showing 'Inbound Authentication - Transport' selected under 'Transport'. In the center, a policy catalog item for 'Inbound Authentication - Transport' is highlighted with a green border. On the right, a 'Policy properties' dialog is open, also titled 'Inbound Authentication - Transport'. This dialog lists four authentication methods with checkboxes: 'Kerberos Token Authentication', 'HTTP Basic Authentication', 'OpenID Authentication', and 'JWT Authentication'. The 'Save' button is visible at the top right of the dialog.

Software AG Training | 4 - 11

Notes:



Exercise 6

- Enforcing Security Policy

Notes:



Inbound Authentication Message Level

Notes:

Message Security

- User credentials and claims are encapsulated in every message using WS-Security specifications to secure messages
 - Any type of security credentials can be used, mostly independent from Transport, as long both the client and service agree
- Advantage:
 - Message security directly encrypts and signs the message, multiple hops don't break security
 - Partial or selective message encryption and signing is allowed
 - Transport independent
 - Wide set of credentials and claims is supported
- Disadvantage:
 - Only supported for SOAP Service
 - May reduce performance compared to transport security

Notes:

Authentication Message Layer - WS-Security Policy Actions

- Applicable only for SOAP APIs
 - Used for a variety of use cases with security
- Inbound Authentication – Message Policies
 - To configure WS-Security actions evaluated by API Gateway
- Outbound Authentication – Message Policies
 - To configure WS-Security actions enforced by native service and added by API Gateway
- API Gateway security definition is required (Keystore, Truststore, ...)



Inbound Authentication - Message

Binding Assertion

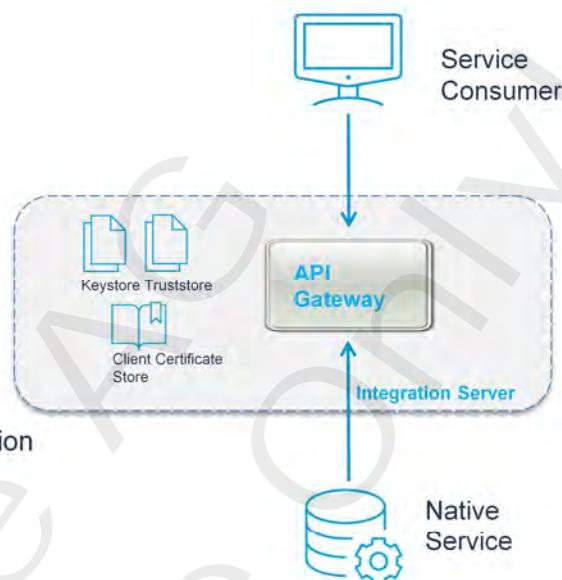
- Require Encryption
- + Add require encryption
- Require Signature
- + Add require signature
- Token Assertions
- Require X.509 Certificate
- Require WSS Username token
- Require Kerberos Token
- Require SAML Token
- Custom Token Assertion
- + Add
- Require Timestamp

Software AG Training | 4 - 15

Notes:

Configuration of WSS Security Policies

- Separate sections
 - Require Encryption to define
 - Encrypted Parts and Elements
 - Require Signature to define
 - Signed Elements and Parts
 - Token Assertions to check
 - Require X.509 Certificate / Require WSS Username Token /
 - Require Kerberos Token /
Require SAML Token / Custom Token Assertion
 - Check box for Require Timestamps
- Covered in a separate chapter: P12 - Advanced Security Processing



Software AG Training | 4 - 16

Notes:



Exercise 7

- Using a WSS Security Token

Notes:



Authorize User

Notes:

The screenshot shows the Software AG Policy catalog interface. On the left, there's a navigation pane with sections like Threat protection, Transport, Identify & Access, and Identify & Authorize Application. Under Identify & Access, there are sub-options for Inbound Authentication - Transport and Inbound Authentication - Message. A policy named 'Inbound Authentication - Transport' is selected. On the right, a detailed view of the 'Authorize User' policy is shown. It includes fields for 'List of Users' and 'List of Groups', both with search input fields. The 'Open in full-screen' button is also visible.

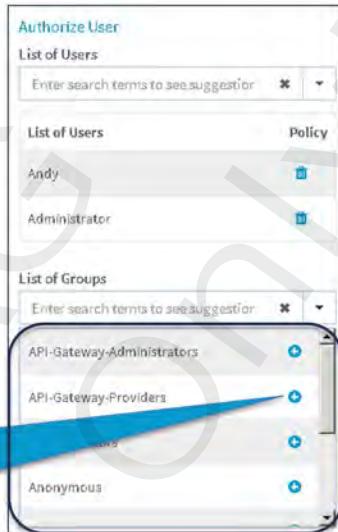
- Authorize User
 - Has dependent authentication action:
 - Inbound Authentication - Transport (default) or
 - Inbound Authentication – Message for SOAP APIs
 - Policy action will be automatically moved into applied policy actions
 - In case of SOAP APIs a dialog box will ask to associate the Inbound Authentication - Transport

Software AG Training | 4 - 19

Notes:

Authorize User Configuration

- List of Users
 - Lists all users defined in Integration Server User Management
 - Administrator, Developer, John, ...
- List of Groups
 - Lists all users defined in Integration Server User Management
 - API-Gateway-Administrators, API-Gateway-Providers, Administrators, Everybody ...



A screenshot of the 'Authorize User' configuration interface. It shows two sections: 'List of Users' and 'List of Groups'. The 'List of Users' section contains entries for 'Andy' and 'Administrator'. The 'List of Groups' section contains entries for 'API-Gateway-Administrators', 'API-Gateway-Providers', 'Administrators', and 'Anonymous'. A blue callout box points to the 'List of Groups' section with the text 'Hit + sign to select a group from the list'.

Software AG Training | 4 - 20

Notes:



5

Consumer Management

Notes:

Objectives

At the end of this chapter you ...

- Understand the Concept of Applications
- Know the difference between global Consumers and registered Consumers
- Know different identifiers in Application

Notes:

Chapter Contents

- Consumers
- Registered Application
- API Key as special kind of Consumer
- Approval Workflow within Application Management

Notes:

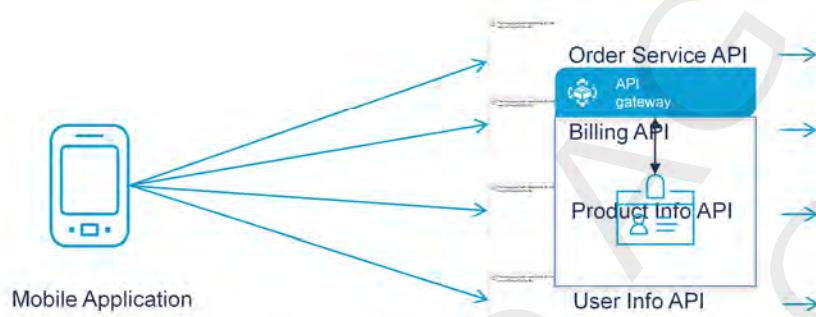


Consumers

Notes:

Application

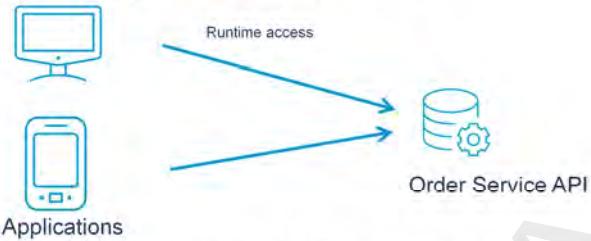
- API Gateway helps to restrict API access with different Identification criteria



Software AG Training | 5 - 5

Notes:

Benefits of Applications in API Gateway

- Control Access to APIs
 - consuming (invoking) service at runtime
- Monitor an API for violation of SLA for a specified application
- Identify the consumer application that a logged transaction is associated with



Account Schema

Software AG Training | 5 - 6

Notes:

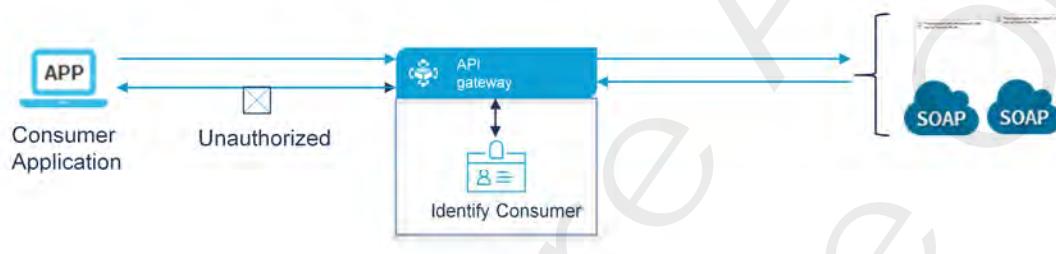
Consumer Applications

- Consumer Applications
 - provide a mechanism by which a run-time consumer can be identified
 - contain a list of consumer identifiers, which will be used for identification at run-time
 - consumer submits the consumer request to the API Gateway
- Consumer Identifiers
 - are defined in a consumer application
 - are used by run-time policies to identify or authorize consumers at run-time,
 - Application without a Policy action requesting consumer authorization has no effect at runtime
- Policy Enforcement Point (API Gateway)
 - Hosts the Application
 - tries to map the consumer's identifier to an identifier in a consumer application
 - If configured in the policy action verifies that the application is mapped to the API

Notes:

Applications in API Gateway

- Relationship between asset and consumer
 - Design Time action by API Provider
 - create Application and define Identification Criteria
 - Realize mapping between Application and APIs (optional) => registered application
 - Run Time action by API Gateway
 - Monitor and control access to authorized and identified consumers
 - Application defines precise identifiers, by which API Gateway can recognize messages from a particular consumer at run-time



Notes:

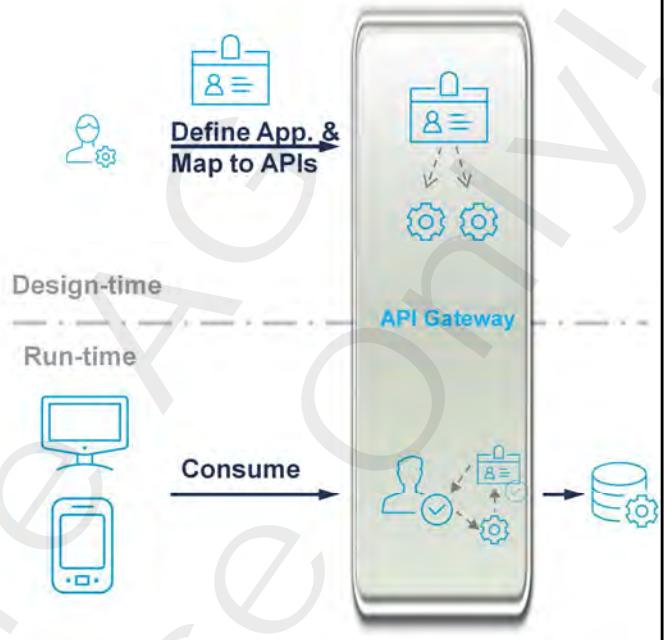
Identification and Authorization by Application

- Access to the API is based on an Application in API Gateway
 - Available for REST and SOAP APIs
 - Provides identification and authorization
- Global application
 - Consumer has to be identified as a member of an application based on Application Identifier definition in the application
 - Just Identification and authorization based on membership to any application
- Registered application
 - Consumer has to be identified as a member of an Application
 - API which the consumer wants to access has to be registered to the application
 - Identification and authorization for a specific API

Identity and Access management, in short, IAM , is identifying consumer of a API request. API clients use variety of identifiers to get identified and API Gateway is capable of identifying consumer using IP Address of client, Basic Auth credentials, Hostname, API key, OAuth2 Token, SSL Certificate, Any attribute or element in the XML message using Xpath expression, WS Security Username and X.509 certificate. Consumer can be identified with any one of the identifiers or with a set of identifiers using AND OR conditions. Policy can be configured to define the lookup condition to be either global or registered applications.

Registered Consumer Identification and Authorization

- API provider in API Gateway can enforce for a service
 - Consumer Identification
 - Consumer Validation based on an application
- Application can be defined in API Gateway
 - Based on a specific Identifier
- Application can be mapped to API(s)
 - 1 Application => n APIs
 - 1 API => n Applications



Software AG Training | 5 - 10

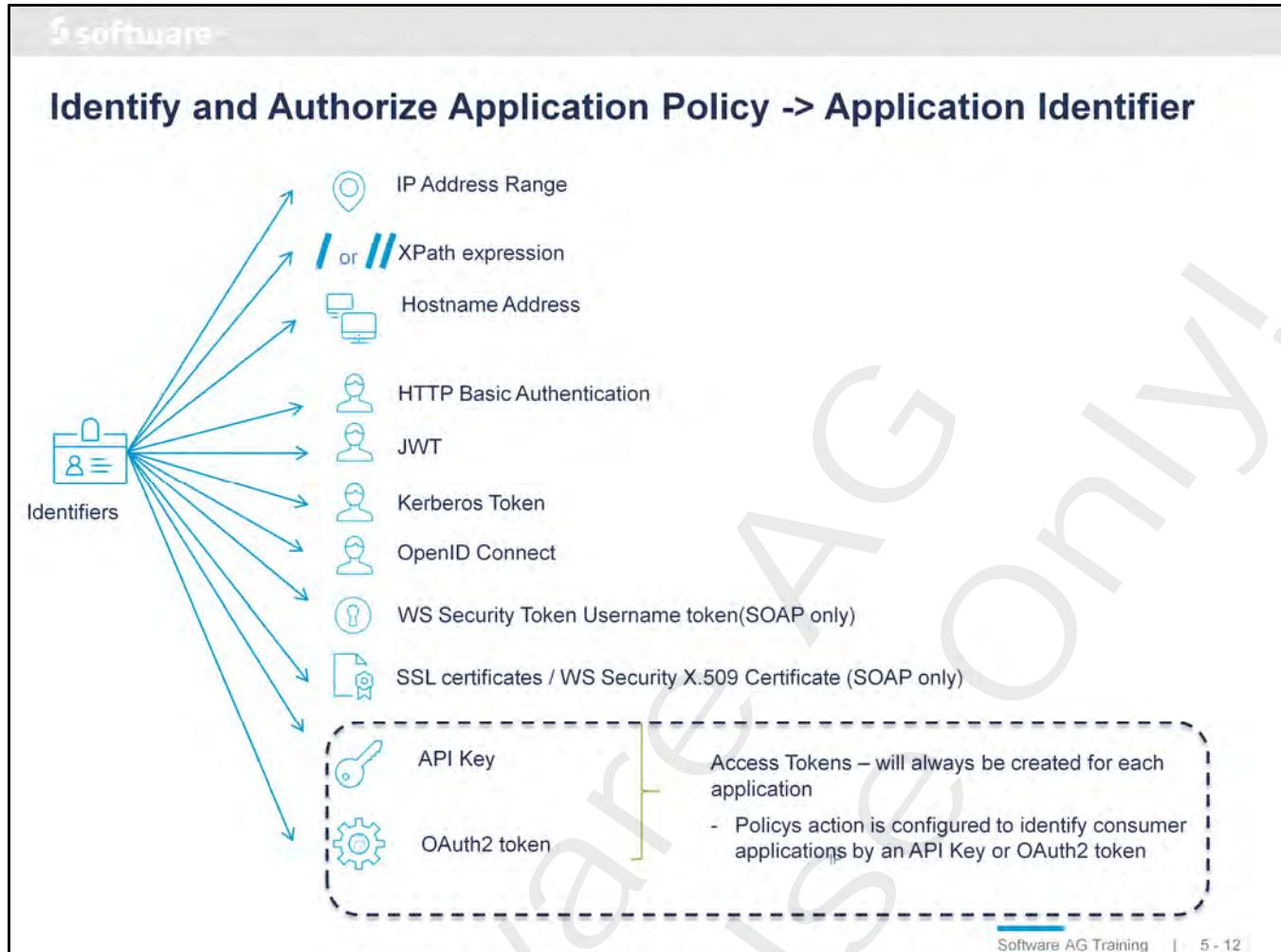
Notes:

Definition of Application Identifiers

- Basic Identifiers
 - IP Address range 
 - HTTP Basic Authentication, Kerberos Token, OpenID, JWT 
 - Hostname 
 - XPath expression  or 
 - WS Security Token Username token(SOAP only) 
- Access Tokens
 - API Key 
 - OAuth2 Token 
- Client Certificates
 - SSL Certificate 
 - WS Security X.509 Certificate (SOAP only) 

Software AG Training | 5 - 11

Identity and Access management, in short, IAM , is identifying consumer of a API request. API clients use variety of identifiers to get identified and API Gateway is capable of identifying consumer using IP Address of client, Basic Auth credentials, Hostname, API key, OAuth2 Token, SSL Certificate, Any attribute or element in the XML message using Xpath expression, WS Security Username and X.509 certificate. Consumer can be identified with any one of the identifiers or with a set of identifiers using AND OR conditions. Policy can be configured to define the lookup condition to be either global or registered applications.



Notes:

Identify & Access: Identify & Authorize Application

Identify & Authorize Application

- Identify & Authenticate Consumer
 - Based on Identification Type
- Authorize Consumer
 - Based on Application Lookup Condition:
 - Global or registered application

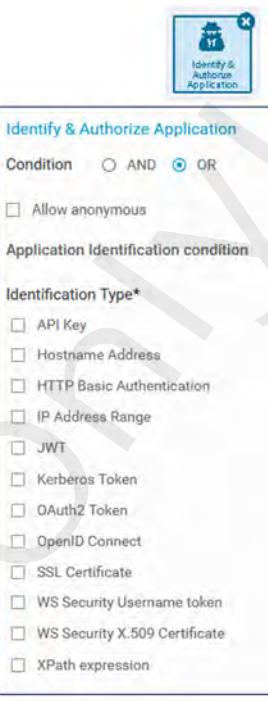
Software AG Training | 5 - 13

Notes:

Identification Criteria

- Criteria Can be combined
 - And / Or

- Identification Types:
 - Hostname Address
 - IP Address Range
 - Identify consumers based on one or more specific IP addresses, or by a special range of IP addresses
 - XPath expression
 - Custom identification token (XPath expression on the SOAP or XML message/request)
 - Access Tokens (API Key, OAuth2 Token)
 - Transport Layer
 - HTTP Basic Authentication, SSL Certificate, Kerberos Token
 - Message Layer - (only for SOAP)
 - WS Security Username token, WS Security X.509 Certificate



Notes:

Application Lookup Condition

- Use Input Parameter **Application Lookup Condition** to define the list of consumers against which the consumer identifier should be validated
 - Global applications
 - Registered applications
- This parameter is not available for access tokens:
 - API Key
 - OAuth2 Token

The screenshot shows the Software AG Policy Catalog interface. On the left, there's a navigation tree with categories like Threat protection, Transport, Identify & Access, Request Processing, Routing, Traffic Monitoring, Response Processing, and Error Handling. In the center, a policy flow diagram is displayed. A green arrow points from the 'Identify & Authorize Application' step to the 'Application Lookup condition' dropdown in the policy properties panel on the right. The dropdown menu is open, showing three options: 'Registered applications' (which is selected and highlighted in blue), 'Global applications', and 'Other consumer types'. Other visible policy properties include 'Condition AND OR', 'Allow anonymous', 'Identification Type' (with 'HTTP Basic Authentication' checked), and 'Kerberos Token'.

Notes:

The screenshot shows two configurations of the 'Identify & Authorize Application' screen. On the left, the 'Condition' is set to 'AND'. Under 'Identification Type*', 'HTTP Basic Authentication' is selected. In the 'Application Lookup condition' section, 'Global applications' is chosen from a dropdown menu. A blue box highlights this selection. A blue arrow labeled 'save' points to the right configuration. In the right configuration, the 'Condition' is now 'OR'. The 'HTTP Basic Authentication' checkbox is checked. The 'Allow anonymous' checkbox is checked and set to 'False'. The 'Global applications' option is also present in the dropdown menu.

Software AG Training | 5 - 16

Notes:

API Key / OAuth2 Token

- API Gateway extracts API Key / OAuth2 token from client request
- The identity is always verified against the list of consumer applications who are mapped to the specified API
 - Application Identifiers must contain API Keys /OAuth2 Tokens

The diagram illustrates the configuration of 'Identity & Authorize Application' for two different identification types: API Key and OAuth2 Token. Each configuration includes a 'save' button and a detailed view of the condition settings on the right.

Configuration 1: API Key

- Condition:** OR
- Application Identification condition:**
 - Application Lookup condition
 - Registered applications
 - Identification Type: API Key
 - Allow anonymous: False

Configuration 2: OAuth2 Token

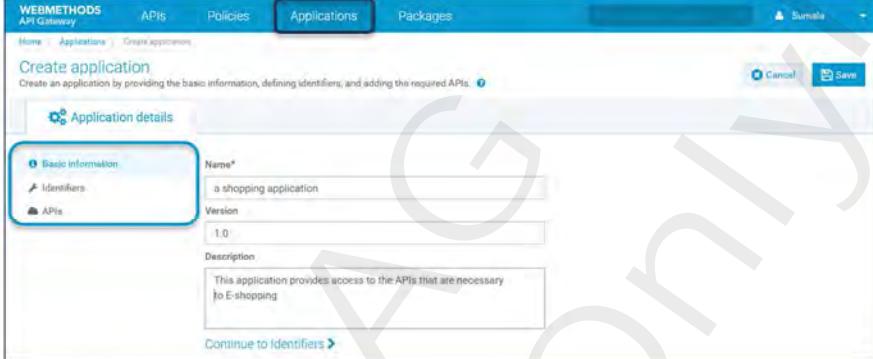
- Condition:** OR
- Application Identification condition:**
 - Application Lookup condition
 - Registered applications
 - Identification Type: OAuth2 Token
 - Allow anonymous: False

Software AG Training | 5 - 17

Notes:

Create Consumer Application

- Applications can be defined
 - Global Header Menu
 - API Menu
- Application Definition
 - Basic Information
 - Identifiers
 - Select Identifier Type
 - Define values
 - APIs (Optional)
 - Assign APIs
 - Can also be assigned from the APIs detail page
- Application State/Mode
 - Edit-Mode
 - Saved-Mode
 - No activate action is needed



Software AG Training | 5 - 18

Notes:

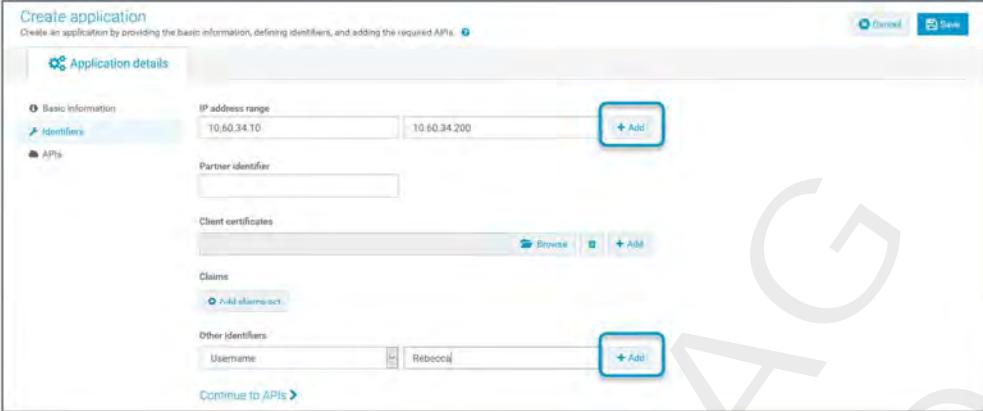
Application – Definition of Identifiers

The diagram illustrates the various types of identifiers used in API management:

- IP Address Range
 - Define a range of IP addresses
- Partner Identifier
 - Provides 3rd partner's identity
- Client Certificate
 - Upload client certificate
- Claims
 - Set of claims for JWT and OpenID clients
 - Name-value pair defines unique identifying information
- Other Identifiers
 - Hostname, Token, Username, WS-Security username, XPath

Software AG Training | 5 - 19

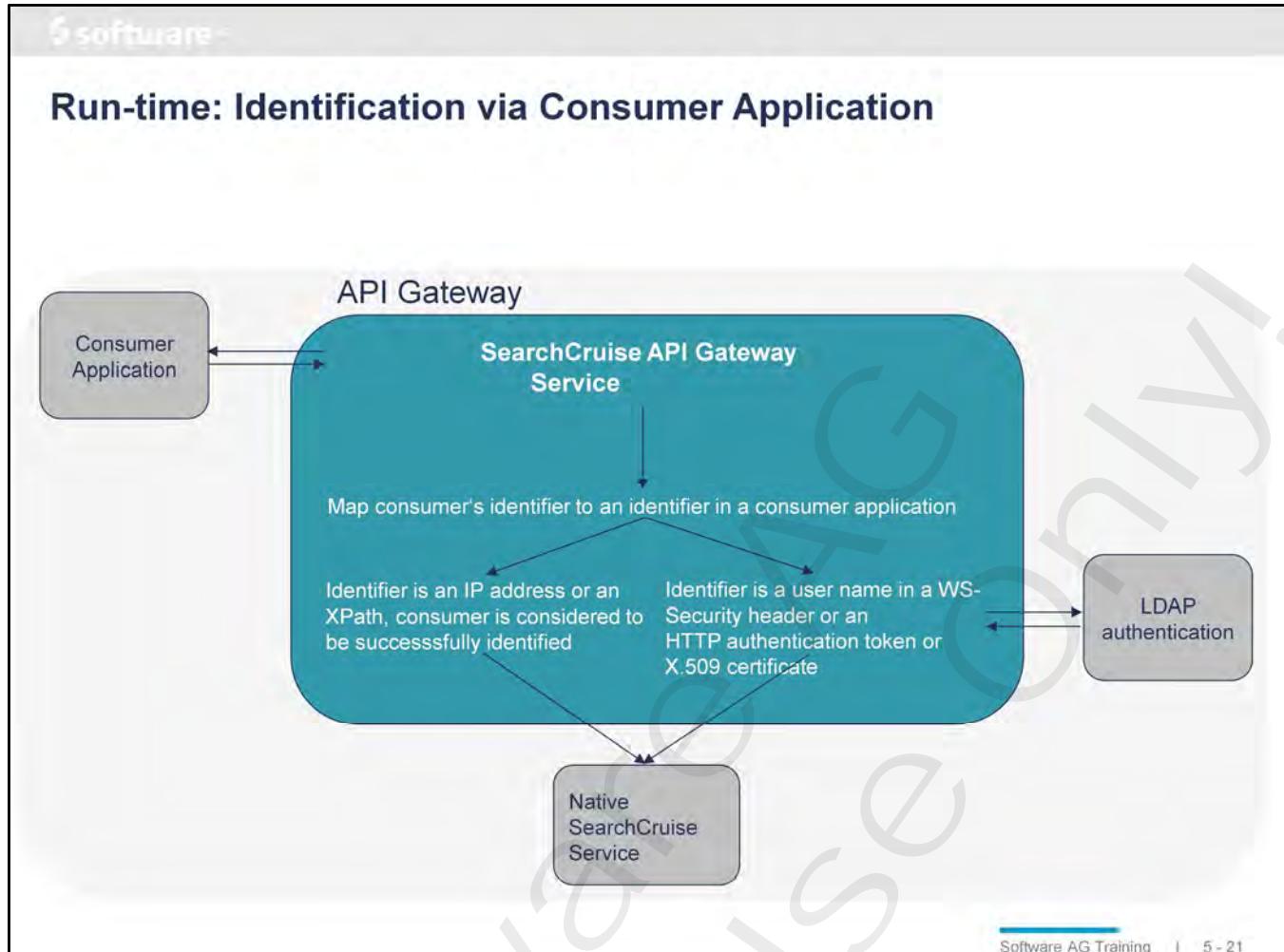
Notes:



- API Key and OAuth2 Token will be generated when creating the application
- Definition of multiple values of the same Identifiers Type is valid
- Multiple Identifiers Types can be defined
- When API Gateway finds ONE identifier matching ONE value in the application, the access will be granted

Software AG Training | 5 - 20

Notes:



Notes:



Registered Application

Notes:

The screenshot displays three separate application management interfaces:

- MyFirstApplication:** Shows basic details like Application ID, Name, and Status. It includes tabs for Identifiers, Policies, Applications, and Analytics.
- SearchCruise:** Shows API details, Policies, Applications, and Analytics. A dropdown menu under "Identify & Authorize Application" is highlighted, showing options: Registered applications (selected), Global applications, and API applications.
- SearchCruise (Selected applications):** Shows a list of selected applications. One application, "MyFirstApplication", is highlighted with a blue border. A blue curved arrow points from the "MyFirstApplication" entry in the list back to the "Registered applications" option in the dropdown menu above.

Notes:

Create application
Create an application by providing the basic information, defining identifiers, and adding the required APIs.

Basic Information

- Identifiers**
 - APIs** (highlighted)
 - Client certificates
 - Other identifiers
- IP address range
- Username
- Resources

Find APIs
Type a keyword...

Name	Description	Version
Swagger	This is a sample service Petstore server. You can find out more about Swagger at http://swagger.io .	1.0.0
AirportInfo	This is a sample service providing Airport Information	1.0
SearchCruise	This service enables users to search for tours offered by SAGTours	1.0

Selected APIs

Name	Description	Version
AirportInfo	This is a sample service providing Airport Information	1.0

Software AG Training | 5 - 24

Notes:

Different Type of Consumers

- Identifiers for identification and authorization, specified by API Provider
 - A list of precise consumer identifiers
- Access Tokens as generated by API Gateway (later in this lesson)
 - API Key
 - OAuth2 Credentials

Software AG Training | 5 - 25

Notice that this is the 'Consumers' page, selected from the left-hand menu on the Mediator panel. Displays each Consumer App, with identifying tokens, IP ranges or client certificate. Also creation and last modification dates.

The 'Synch All Consumers' button is shown. Operation was described on a previous slide.

Levels of Authorization Rules for Consumer at Runtime

The diagram illustrates the three levels of authorization rules for a consumer at runtime:

- Inbound Authentication - Transport:**
 - Require Kerberos Token (unchecked)
 - Require HTTP Basic Authentication (checked)
- Authorize User:**
 - List of Users: Andy, Administrator
 - Policy: Two blue icons representing policies.
- Identify & Authorize Application:**
 - Condition: AND (radio button selected)
 - Allow anonymous (unchecked)
 - Application Identification condition: Identification Type* (API Key, Hostname Address, checked: HTTP Basic Authentication)
 - Application Lookup condition: Registered applications (selected), Registered applications, Global applications

Blue arrows point from each section to its corresponding icon in the center of the slide.

- Inbound Authentication:**
 - Just Authentication
- Authorize User:**
 - User has to provide authentication
 - API Gateway will check that user is listed as authorized user
- Identify and Authorization Application:**
 - User has to be identified in an application
 - Application has to be registered for the API (Registered applications option)

Notes:



Exercise 8

- Creating a Consumer Application

Notes:

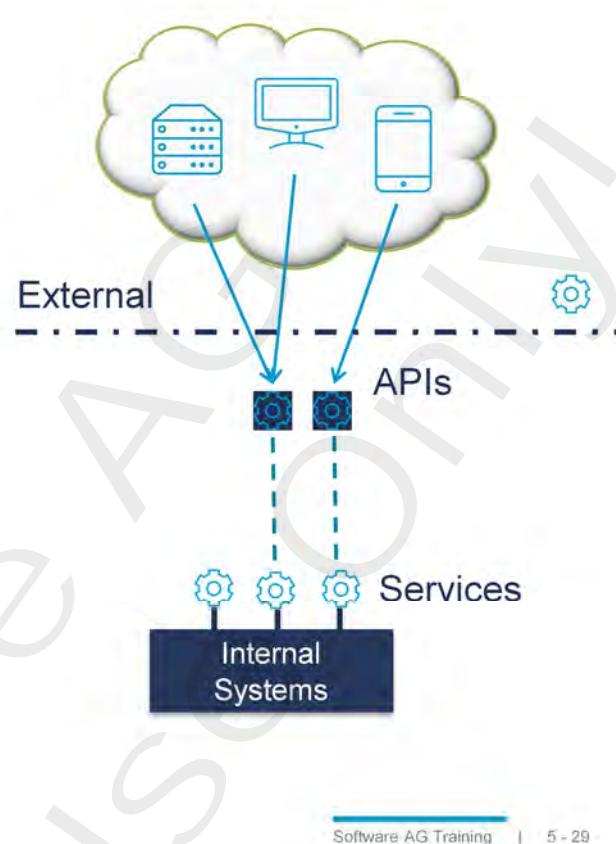


API Key as special type of Consumer

Notes:

Securing your APIs with API Keys

- Challenge
 - Manage external APIs where consumers are unknown
- API Key = Information passed in when calling an API
 - to identify
 - Calling program
 - to track and control how the API is being used
 - prevent malicious use or abuse



Software AG Training | 5 - 29

An Application Programming Interface Key is generated by CentraSite to identify the API, its provider or its consumer.

The API key acts as a unique identifier and a secret token for authentication. Generally it will have a set of access rights on the API.

API Key Definition

- What's an API Key?
 - Generated unique identifier
 - Used to identify the calling consumer application
 - Passed along with every request
 - Can be combined with other security & authentication criteria
- Send from Client to API
 - As HTTP Header or
 - As query parameter

GET /api/v1/projects
X-Gateway-APIKey:
9aae75efd6b3470db8abd15b61c55cef

```
<script type="text/javascript"  
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY"> </script>
```

<http://openweathermap.org/data/2.3/forecast/city?id=524901&APPID=111111111>

Software AG Training | 5 - 30

Notes:

Application – Access Token Management

- Access Tokens are generated by API Gateway when creating the application
- Access tokens can NOT be deleted
- API Key
 - Can be regenerated
- OAuth2 Credentials
 - Can be refreshed

The screenshot shows the 'My Order System' application details page. On the left, there's a sidebar with 'Application Details' and links for 'Basic information', 'Identifiers', 'Access tokens' (which is highlighted), and 'APIs'. The main content area has two sections: 'Access tokens' and 'OAuth2 Credentials'. The 'Access tokens' section contains a table with columns 'API key' and 'API access key'. The first row shows an API key with value 'dfea7e40-0fef-486c-9cc0-72237b7c24ff' and an access key with value '3a2cad07-21d3-41cc-b02a-5874dc9fa5b9'. Both rows have a blue 'C' icon in a box to their right. A blue arrow points from the 'C' icon next to the first API key to the 'C' icon next to the 'API key' header in the 'Access tokens' section of the main content.

Notes:

Activate API Key as Access Control

- Access control by Application has to be defined within API policy action definition
 - Identify & Authenticate Consumer
 - API Key
 - No Application Lookup Definition is provided
- Access control with API Key
 - Always enforces a consumer to send an API Key which is listed in an Registered Application

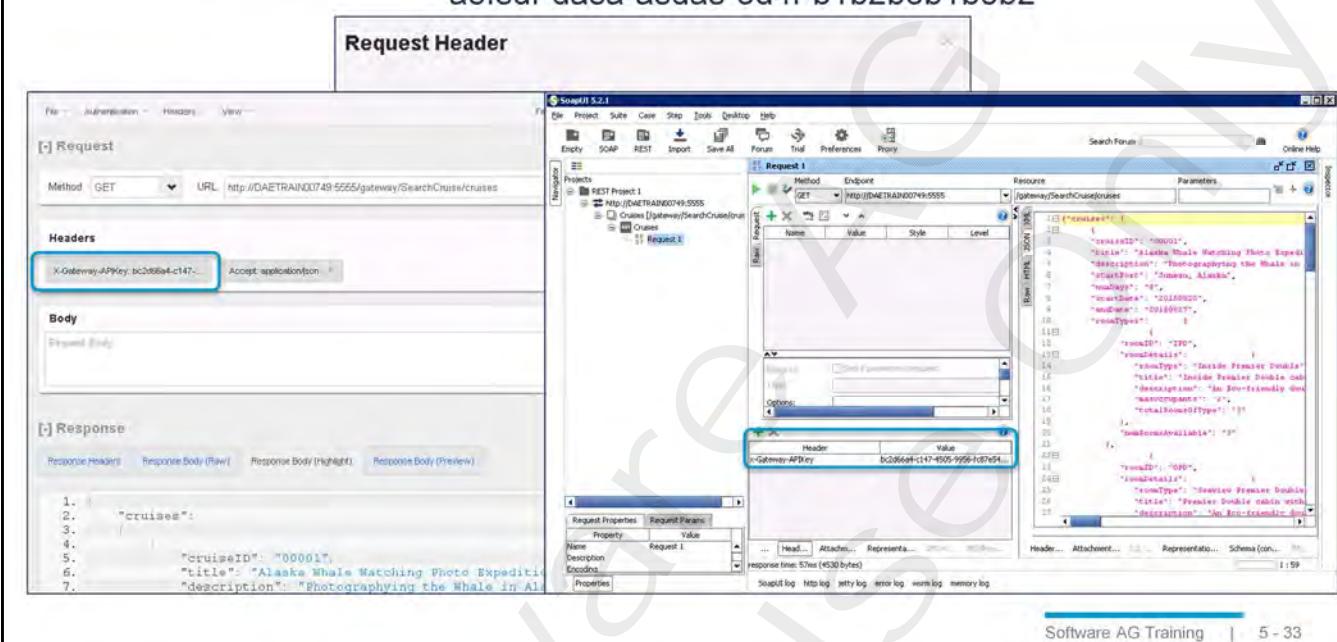
The diagram illustrates the configuration of an API policy action. On the left, a detailed view shows the 'Identify & Authorize Application' configuration with 'Condition' set to 'OR'. It includes an 'Allow anonymous' checkbox and an 'Application Identification condition' section. Within this section, 'Identification Type' is set to 'API Key', and other options like 'Hostname/Address', 'IP-Address Range', and 'OAuth2 Token' are listed but not selected. A large blue arrow points to a simplified configuration on the right, where the 'Application Identification condition' is directly set to 'Registered applications', bypassing the intermediate step of selecting 'API Key'.

Software AG Training | 5 - 32

Notes:

Consume API using API Key in HTTP Header

- HTTP Header corresponds to an array of header names
 - HTTP Header Name: x-Gateway-APIKey
 - HTTP Header Value: <x-Gateway-API Key>
a3fsdf-dasa-asdas-3d4f-b1b2b3b1b3b2



Notes:

The screenshot shows the 'MyFirstApplication' application details page. On the left, there's a sidebar with tabs: 'Application Details' (selected), 'Basic Information', 'Identifiers', 'Access tokens', and 'APIs'. The main content area has tabs: 'MyFirstApplication' (selected), 'Create an application by providing the basic information, defining identifiers, and adding the required APIs.', and 'Edit'. The 'Edit' tab is active, showing the 'Application details' form. In this form, the 'OAuth2 Credentials' tab is selected. The form fields include:

- Type: Confidential
- Token lifetime:
 - Unlimited
 - Limited
- Token Lifetime (in seconds): 3600
- Token refresh limit:
 - Limited
 - Unlimited
- Token refresh limit: 0
- Redirect URIs: A list with a '+ Add' button.

At the bottom right of the edit form are 'Cancel' and 'Save' buttons. The top right of the main content area has 'Edit' and 'Save' buttons. The bottom right of the page shows 'Software AG Training | 5 - 34'.

Notes:



Approval Workflow within Application Management

Notes:

Approval within Application Management

- Approval process can be configured to enforce an approval
 - Create
 - Register
 - Update
- Different approvers can be defined for each process (create, register, update)

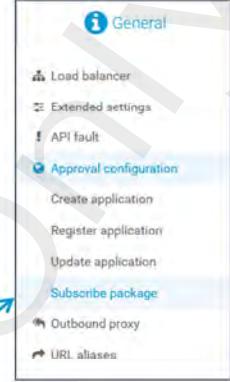


Software AG Training | 5 - 36

Notes:

Approval Options

- Administrator privilege is needed to to Approval configuration
 - General administration configuration
- After enabling the Approval functionality following Definition is possible
 - Approvers / Approvers group
 - Access Profiles as a drop down list
 - Approval Mode
 - Anyone (as of now)
 - Approval initiate request template of email to be sent to approver
 - Request approved template of email to be sent to requester
 - Request rejected template of email to be sent to requester
 - All of these configurations include subject and content of the mail
- Subscribe Package is an additional option/process to enforce approval



A screenshot of the Software AG Administration interface. On the left is a navigation tree with the following items:

- Load balancer
- Extended settings
- API fault
- Approval configuration** (highlighted in blue)
- Create application
- Register application
- Update application
- Subscribe package
- Outbound proxy
- URL aliases

A blue arrow points from the text "Subscribe Package is an additional option/process to enforce approval" to the "Subscribe package" item in the navigation tree.

Software AG Training | 5 - 37

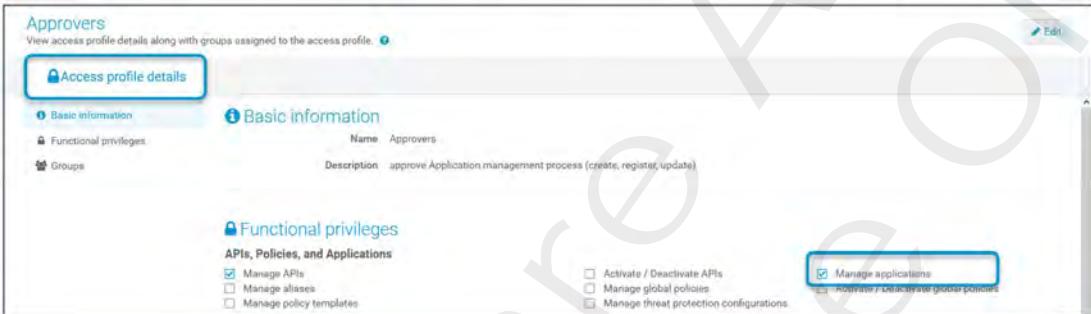
Notes:

The screenshot shows the 'Administration' section of the API Gateway interface. On the left, a sidebar lists various configuration options like General, Security, Load balancer, Extended settings, API fault, and Approval configuration. The Approval configuration option is selected. The main panel is titled 'Create application' and 'Configure the approval workflow for creating a new application'. It includes an 'Enable' switch (which is turned on), a dropdown for 'Approvers' (set to 'Administrators'), and a dropdown for 'Approved by' (set to 'Anyone'). A large blue callout bubble points to the 'Approvers' dropdown with the text 'Access profile of approvers'. Another blue callout bubble points to the 'Approved by' dropdown with the text 'Approval mode'. On the left side of the main panel, there is a note: 'email configuration to be sent to approver / requestor'. At the bottom of the panel are 'Cancel' and 'Save' buttons. The footer of the page reads 'Software AG Training | 5 - 38'.

Notes:

Requested Function Privileges for Approving an Application

- Approvers in Approval Workflow must be a
 - User associated with the selected access profile which includes **Manage Applications**
 - User associated with the **Administrator** access profile
- The function privilege **Manage Application** does not include modify permission on applications created by another user
 - Only the Administrator access profile gives permission to modify any application



Software AG Training | 5 - 39

Notes:

Application Creation with Approval Enabled

- Creating an registered application for which the workflows **Create application** and **Register application** are enabled enforces 2 different approval processes,
 - First the Create application process must be approved (only this process is listed in pending requests)
 - After the creation of the application is approved/created the 2nd process Register application is listed in pending requests
 - The approval configuration for both workflows can have different approvers

Software AG Training | 5 - 40

Notes:

Managing Applications Summary

- API Gateway Provider creates the API in API Gateway
 - creates an Identification and Authenticate Consumer Policy action as part of the API definition, referencing
 - Global Application
 - Registered Application
 - Using corresponding Identification Type mapping to application Identifiers
- API Gateway Provider creates an Application to control access to API
 - defines Identifier values matching API policy definition
 - assigns the API(s) to the application -> registered Application
- API Gateway at runtime tries to find a match on identifiers
 - Global Application criteria => is matched => access to the API is granted
 - Registered Application criteria => find a match of the API in the Application => is matched => access to the API is granted
- API Gateway Provider and API Gateway Administrator can manage applications (default configuration)

Software AG Training | 5 - 41

Notes:



Exercise 9

- Enforcing a Consumer Relationship using API Key

Notes:



6

Policy Management

Notes:

Objectives

At the end of this chapter you ...

- Know how to enable Threat Protection in API Gateway
- Understand how to manage Global Policies and Policy Templates

Notes:

Chapter Contents

- Threat Protection Policies
- Global Policies
- Scope Level Policies
- Policy Templates

Notes:



Threat Protection Policies

Notes:

Exposing APIs to External Clients

- Use case:

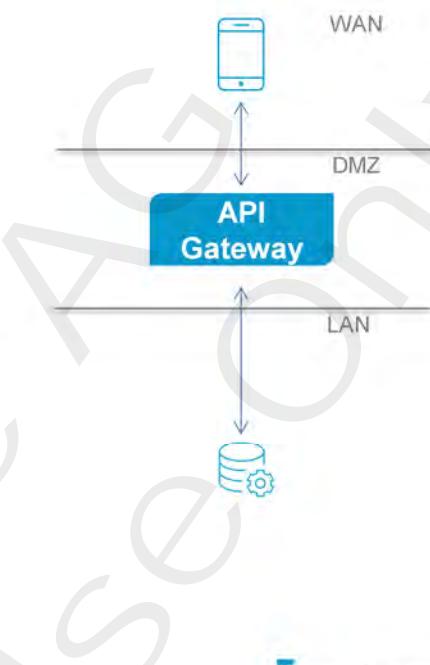
- SOAP / REST API residing in internal applications are behind a firewall
- External clients – mobile apps – are not allowed to communicate with API

- Requirements:

- Protect the APIs with DMZ level protection
- Manage and govern the use of the APIs

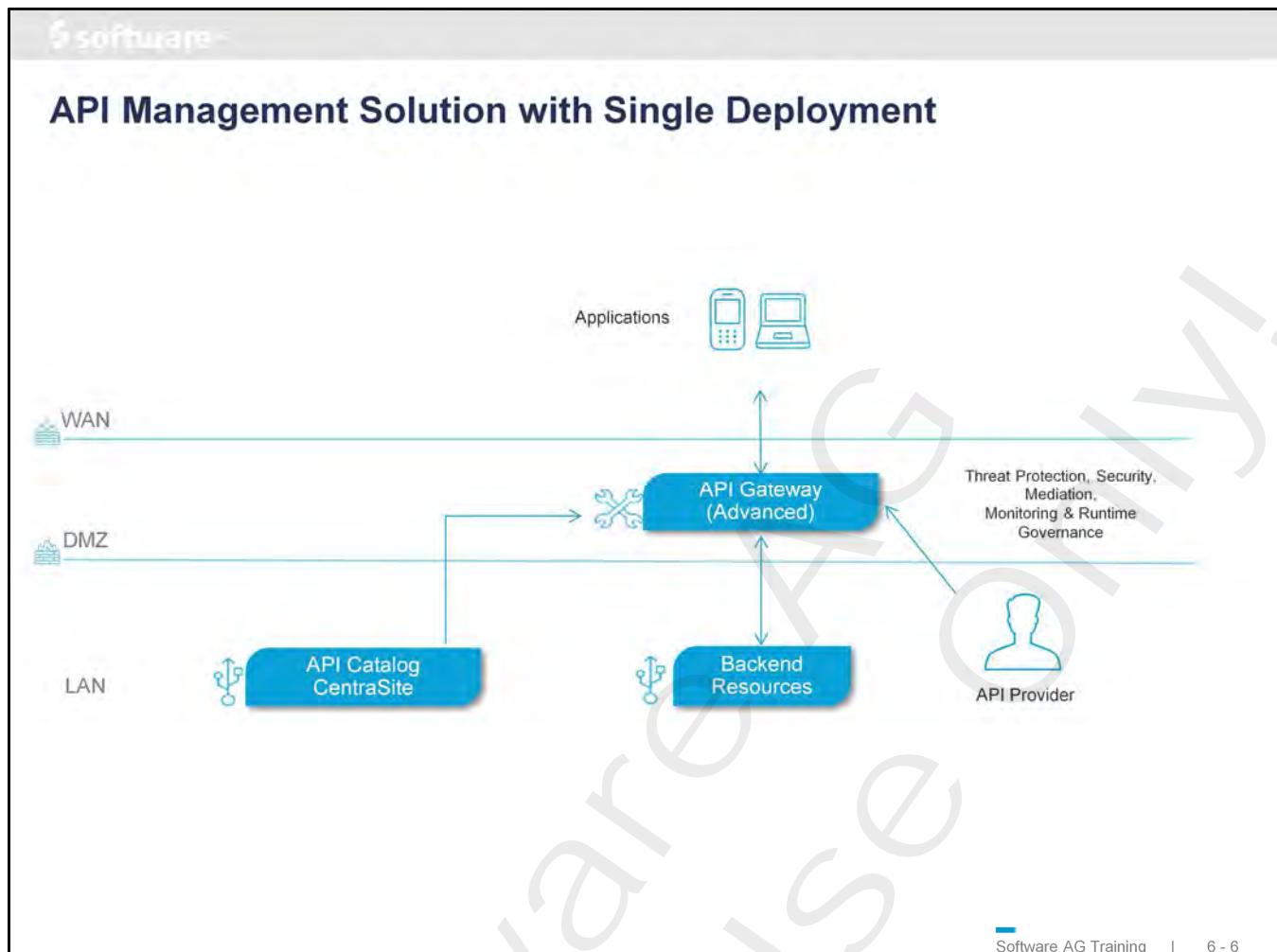
- Solution:

- API Gateway
 - Located in the DMZ
 - Impose Threat protection and security rules on the request

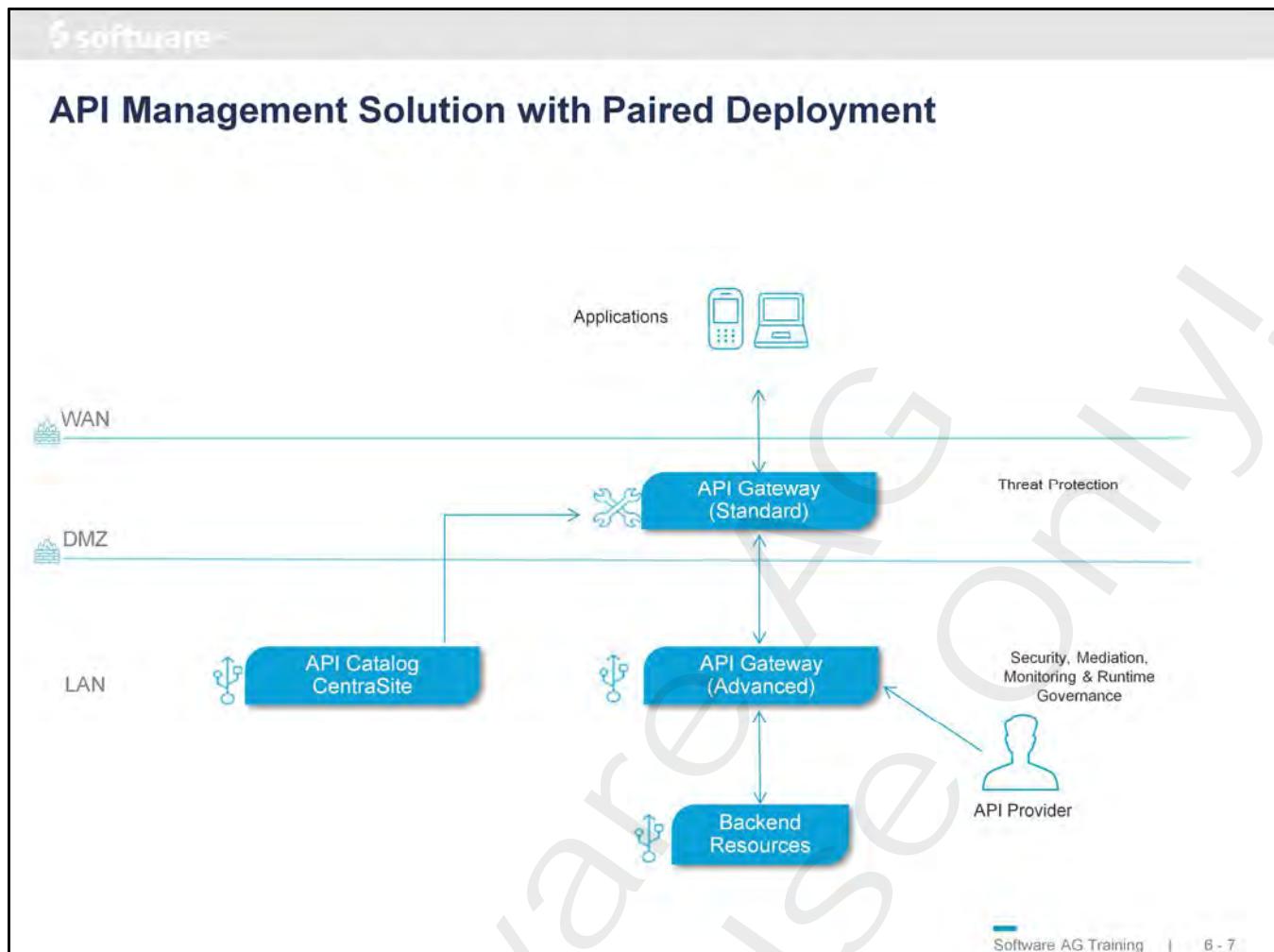


Software AG Training | 6 - 5

Notes:



Notes:



Notes:

API Gateway DMZ-Level Security

- API Gateway Server is placed in the DMZ
 - Imposes Threat Protection rules
 - Imposes Security Rules
 - Forwards the requests to the native service on your internal network
 - In case of a request violates a rule
 - Blocks the request
 - Allows to process and sends an alert
- Threat Protection Rules
 - Controls which requests are forwarded to internal IS
 - Protects against various kinds of threats and attacks
 - Denial of Service (DoS) global or based on IP addresses)
 - Trusted IPs (white-list of IP addresses)
 - Filtering:
 - Message Size, Mobile Application & Devices, SQL Injection
 - Antivirus Scan
 - Custom

Notes:

Deployment Scenario: Reverse Invoke



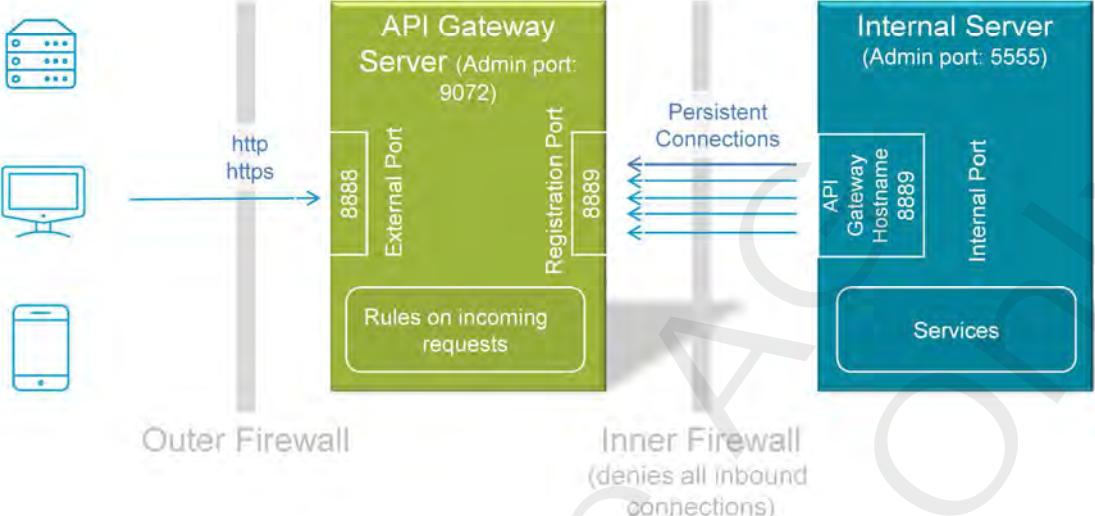
- External client
 - sends request to API Gateway
 - API Gateway
 - Collects client information
 - Evaluates the request against any threat protection rules and policies
 - Passes requests to internal Integration Server
 - Integration Server
 - Processes request and sends response back to API Gateway
 - API Gateway
 - Passes the response back to client

Reverse Invoke Communication

Software AG Training | 6 - 9

Notes:

API Gateway Configuration: Reverse Invoke



- Communication is always initiated from the internal server to the API Gateway
 - No firewall ports need to be opened that could lead to violation of security guidelines

Software AG Training | 6 - 10

Notes:

Ports Configuration

- Configuration in API Gateway > Security section for API Administrators
 - Definition and Enablement of the ports
 - Ports are created in Integration Server (IS Administration UI)
- The webMethods API Gateway external port and registration port work as a pair
 - One port is not functional without the other
- For the Reverse Invoke scenario, the internal port configuration is done on the ESB Server which listens to the **Registration** port
- If clustering is used the Load Balancer should be configured to use the external port

Notes:

The screenshot shows the 'Administration' section of the webMethods API Gateway. It includes three main panels:

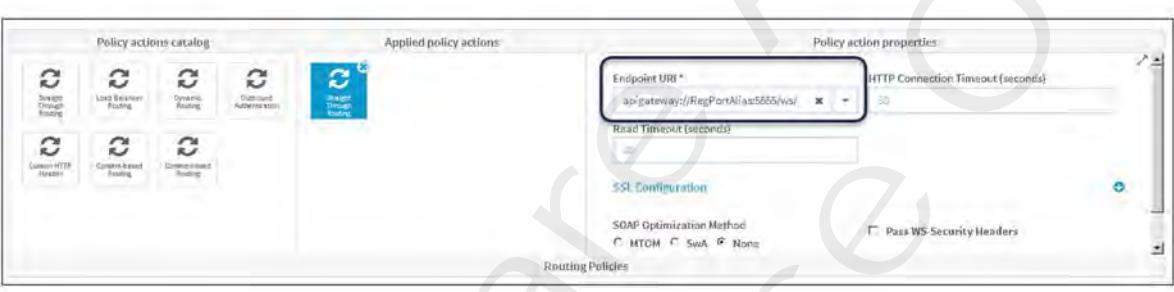
- Left Panel:** Shows the 'Ports' configuration screen where a new port is being added. A blue callout points to the 'Type' dropdown which has 'API Gateway external' selected.
- Middle Panel:** Shows the 'API gateway external listener configuration' for port 8080. A blue callout points to the 'Protocol' dropdown which has 'HTTP' selected.
- Right Panel:** Shows the 'Ports' table with several entries. A blue callout points to the 'Enabled' column for the 'Registration' port, indicating it is enabled.

Software AG Training | 6 - 12

Notes:

Reverse Invoke: API Routing Endpoint Configuration

- Scenario A:
- Internal Server is a webMethods Integration Server
 - Routing endpoint of API created on API Gateway in DMZ has to use the Registration Port Alias name as defined in the external port configuration
 - `apigateway://{{REG_PORT_ALIAS}}/rest/api/resource`
 - on the internal Integration Server the Internal Server Port Configuration has to be set to
 - API Gateway Registration Port



The screenshot shows the 'Policy actions catalog' interface. On the left, there is a grid of icons representing different policy actions: 'Direct Through Routing', 'Load Balancer Routing', 'Dynamic Routing', 'Customized Automation', 'Custom HTTP Header', 'Content-based Routing', and 'Content-based Routing'. The 'Customized Automation' icon is highlighted with a blue border. On the right, under 'Applied policy actions', there is a single item: 'Reverse Invoke'. Below this, in the 'Policy action properties' section, the 'Endpoint URI' field contains the value 'apigateway://RegPortAlias:5555/va/'. Other fields include 'HTTP Connection Timeout (seconds)' (set to 30), 'Read Timeout (seconds)', 'SSL Configuration', 'SOAP Optimization Method' (set to 'None'), and a checkbox for 'Pass WS Security Headers' which is unchecked.

Software AG Training | 6 - 13

Notes:

Reverse Proxy: API Gateway Configuration

The diagram illustrates a three-tier architecture for API management:

- Internet:** Represented by a smartphone icon.
- DMZ:** Represented by a blue box labeled "API Gateway". It has a double-headed arrow connecting it to the Internet.
- Internal Network:** Represented by a blue box labeled "API Gateway" with a gear icon. It has a double-headed arrow connecting it to the DMZ, and another double-headed arrow connecting it to a database icon.

Annotations below the diagram provide specific details:

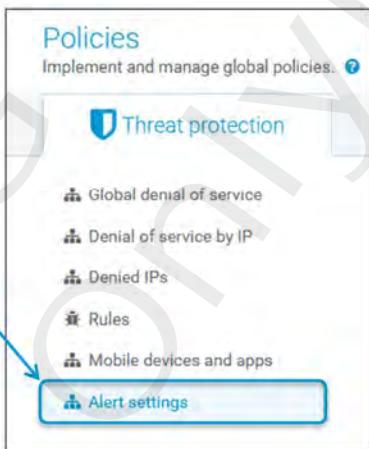
- "Enforcing Threat Protection rules" is indicated between the Internet and the DMZ API Gateway.
- "Enforcing Authentication, Authorization, and Policy enforcements" is indicated between the DMZ API Gateway and the Internal Network API Gateway.

Software AG Training | 6 - 14

Notes:

Threat Protection Alerts

- Policy Violation Configuration
 - Global level
 - Define on the Global Threat Protection Alert Settings
 - Alert Type
 - None (Default)
 - Email to email account
 - Flow service sending Flow Events visible in Threat Protection Dashboard
 - Rule level
 - Define on the Rule Configuration
 - Alert Options
 - None
 - Email
 - Flow Service
 - Overwriting the global alert configuration

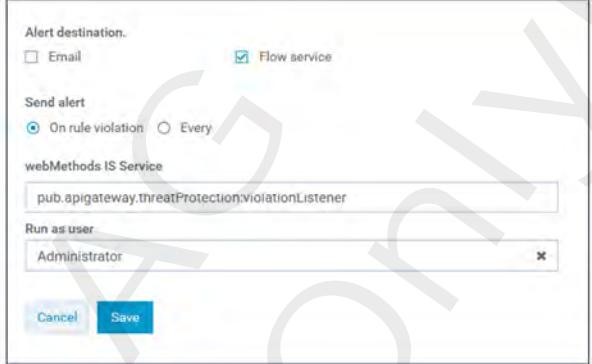


Software AG Training | 6 - 15

Notes:

Alert Type: Flow Service

- Flow Service Configuration
 - Send alert
 - On rule violation
 - Every
 - Specified time interval in minutes
 - Service definition
 - Name of the flow service which has to be invoked
 - Run as user
 - Configures user permission in order to run the service
- Configuration is the same on
 - Global Alert Definition
 - Rules Alert Definition

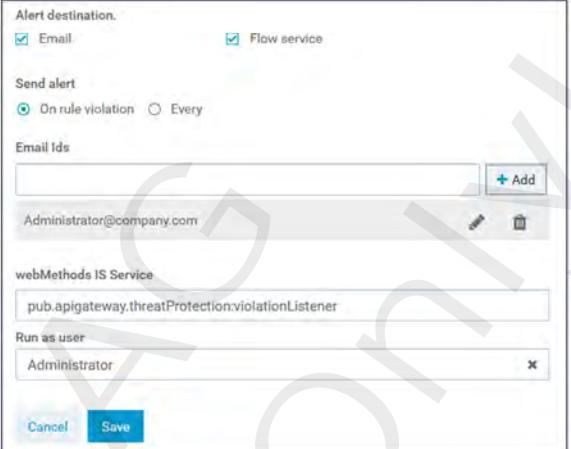


Software AG Training | 6 - 16

Notes:

Alert Type: Email

- Flow Service Configuration
 - Send alert
 - On rule violation
 - Every
 - Specified time interval in minutes
 - Email Ids
 - Any number of email aliases can be configured
 - Any alias receives an alert
- Mail Server Configuration is Integration Server Administration UI
 - Settings > Resources > Email Notification



Software AG Training | 6 - 17

Notes:

Threat protection policies

- Denial of Service
 - Denial of Service (DoS) attacks can be prevented by configuring Denial of Service protection
- Global Denial of Service
 - Global Denial of Service protection is a global entity and it is applied to all the requests irrespective of IP, region or request type
- Denial of Service by IP
 - Denial of Service by IP protection is an IP specific protection and it is applied to all the requests
- Trusted IPs
 - To ensure that requests from trusted servers are not denied, you can configure a white-list of IP addresses so that requests from these IP addresses are always allowed

Software AG Training | 6 - 18

Notes:

Denial of Service (DoS)

- Useless traffic to a service
 - Ping of Death
 - Teardrop attacks
 - Exploit limitations in the TCP/IP protocols
- Threat Protection provides capabilities
 - Denial of Service
 - Global Denial of Service - with excluding IPs
 - Denial of Service by IP
 - Restrict the requests entering the system
 - Prevent the attackers from depleting the resources of the organization

Notes:

The screenshot shows the 'Configuration of Threat protection policies' interface. On the left, there's a sidebar with options: Global denial of service (selected), Denial of service by IP, Denied IPs, Rules, Mobile devices and apps, and Alert settings. The main panel displays configuration for 'Global denial of service':

- Enable: checked
- Maximum requests: 1000
- In (seconds): 10
- Maximum requests in progress: 250
- Block intervals (minutes): 2
- Error message: "Receiving too many requests, Rejecting all requests and entering passive mode !!!"
- Trusted IP addresses: 127.0.0.1

At the bottom are 'Cancel' and 'Save' buttons.

Global entity

- Applied to all requests, irrespective of
 - IP
 - Region
 - Request type

- The client sends 1001 request within 2 minutes
- Client will get an error message as configured and response code 403
- Different client sends additional request within the same time interval
- 2nd client will get the same error

Software AG Training | 6 - 20

Notes:

Configuration of Threat protection policies

IP specific protection
- Applied to all requests,

- The client sends 1001 request within 2 minutes
- Client will get an error message as configured and response code 403
- Different clients invoking the same API from
 - Other IP
 - Trusted IP
- Client will get a successful response

Software AG Training | 6 - 21

Notes:

Software AG Training | 6 - 22

Denied IPs

- List of IP addresses that violates the Denial of service by IP protection
- Administrator checks this list and determines if the request is to be denied
 - Reliable IP
 - Remove from the list

IP address range	Action
10.60.34.31	
10.60.34.152	
10.60.34.74	
10.60.34.17	

Notes:

Threat Protection Rules & Filters

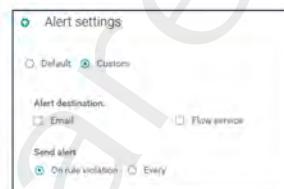
- Rules can be configured to filter malicious request based on different criteria
- Action when a policy is violated:
 - Deny request and Alert
 - Alert
 - Corresponding Error message
- Request Type to apply rule
 - All, Rest, Soap, Invoke, Custom
- Filter Configurations
 - Alert settings
 - Filter

The screenshot shows a software interface for managing threat protection rules. On the left, there's a sidebar with icons for Threat protection, Global policies, and Policy templates. The main area has tabs for 'Threat protection' (selected), 'Global policies', and 'Policy templates'. Under 'Threat protection', there are sections for 'Global denial of service', 'Denial of service by IP', 'Denied IPs' (which is selected and highlighted in blue), 'Mobile devices and apps', and 'Alert settings'. On the right, the 'Rule properties' panel is open, showing a 'Rule name' field with 'MessageSizeRole', an 'Action' dropdown set to 'Deny request and alert', and a 'Request type' dropdown set to 'ALL'. Below these are two sections: 'Available filters' (with options like Alert settings, Message size filter, OAuth filter, etc.) and 'Selected filters' (empty). At the bottom right of the interface, there's a footer with 'Software AG Training | 6 - 23'.

Notes:

Threat protection filters & rules

- Filters can be combined to create rules that are applicable globally as threat protection policies
 - Message Size Filter
 - OAuth Filter
 - Mobile Apps protection filter
 - SQL Injection protection Filter
 - Anti virus Scan Filter
 - JSON threat protection filter
 - XML threat protection filter
 - Custom Filter
- Alert Settings
 - Default
 - Custom configuration



Software AG Training | 6 - 24

Notes:

The screenshot shows the 'Message Size Filter' configuration page. On the left, there's a sidebar with 'Software AG' branding. The main area has a title 'Message Size Filter'. Below it, a bullet point states: 'In cases the native server cannot process very large incoming requests'. The configuration form includes fields for 'Rule name*', 'Action' (set to 'Deny request and alert'), 'Request type' (set to 'ALL'), and 'Description' (a note about restricting incoming requests by message size). It also includes an 'Error message' field ('You have exceeded the message size limit') and a section for 'Alert settings' and 'Message size filter' (with a selected radio button). A toggle switch labeled 'Enable' is turned on. At the bottom right, there's a footer for 'Software AG Training | 6 - 25'.

Notes:

The screenshot shows the 'Rule properties' configuration for an 'OAuth Rule'. The configuration includes:

- Rule name***: OAuth Rule
- Description**: Enforce that the request should contain OAuth credentials.
- Action**: Deny request and alert
- Error message**: OAuth credentials are missing
- Request type**: ALL
- Enable**: Enabled (checkbox checked)
- Require OAuth credentials**: Enabled (checkbox checked)

Software AG Training | 6 - 26

Notes:

Mobile Application Protection Filter

- Disable access for certain mobile application versions on a set of mobile devices
- Filter Option:
 - Ensures that all users use the latest versions of applications
 - Usage of latest security and functional updates
 - Mobile Application Specific Filter
 - Example: restrict incoming request from any mobile device having *Gmail* application version prior to 3.0
 - Mobile Device Specific Filter
 - Example: restrict incoming request from mobile devices other than Samsung having Gmail application older than 3.0
- API Gateway checks for following header fields
 - Mobile-Application-Name
 - Mobile-Application-Version
 - Mobile-Device-Type

Notes:

The screenshot shows the 'Mobile application protection filter' configuration screen. At the top, there are tabs for Threat protection, Global policies, and Policy templates. On the left, a sidebar lists Threat protection (Global denial of service, Denial of service by IP, Denied IPs, Rules), Global devices and apps (Mobile devices and apps, Alert settings), and Policy templates. A blue callout bubble points to the 'Mobile devices and apps' section with the instruction: 'Provide list of - Mobile device type - Mobile application details'. A red arrow points from the 'Mobile devices and apps' sidebar item down to the 'Mobile application protection filter' configuration area. The configuration area contains two rows of rules. Each row has columns for Device type (Samsung), Mobile application (Gmail), Condition (<), and Mobile app version (3). An 'Add' button is available for each row.

Notes:

SQL Injection (SQLI) Filter

- Common attack vector
 - Using malicious SQL Code for back-end database manipulation
 - To access information which was not intended to be displayed
 - Sensitive company data
 - User lists
 - Private customer data
- Result
 - Unauthorized viewing of user lists
 - Deletion of entire tables
 - Gaining administrative right to the database
- Filter Options
 - Database-Specific SQL Injection Protection
 - Standard SQL Injection Protection

Notes:

The screenshot shows the 'Database Specific SLQ Injection Protection' configuration page. It includes fields for Rule name (Test Rule for SQL injection), Action (Deny request and alert), Request type (ALL), and Error message (Test rule violated). Under 'Alert settings', there is a checked checkbox for 'Database-specific SQL injection protection' and a selected database (ORACLE). Under 'SQL injection protection filter', there is an 'Enable' switch and a checked checkbox for 'Database-specific SQL injection protection'. A 'Parameters' section contains three entries: 'userID', 'name', and 'password', each with edit and delete icons. A large watermark reading 'Internal Use Only!' is overlaid across the entire page.

- Parameter Definitions
 - API Gateway will block the request when the parameter contains an invalid special character for example

http://localhost:6666/gateway/PetStoreAPI/pet?userID=jd&password=test

Software AG Training | 6 - 30

Notes:

The screenshot shows the 'Standard SLQ Injection Protection' rule configuration page. It includes fields for Rule name (Test), Action (Deny request and alert), Description, Error message (Test rule violated), and various protection filters like Database-specific SQL injection protection (off) and Parameters (+ Add). A large watermark 'Internal Use Only' is overlaid across the page.

SQL injection protection filter:

- Enable: **on**
- Database: **NONE**
- Standard SQL injection protection: **on**

If enabled:

- API Gateway will block XML and SOAP Payload messages containing
 - Quotation mark (')
 - Number sign (#)
 - Double hyphen (--)
- If no parameter is defined, all parameters are checked

```
<Employee><ID>1245</ID>
<NAME>Albu'm name</NAME>
<DESIGNATION>SS#E</DESIGNATION>
<COUNTRY>USA--</COUNTRY>
<DOJ>2014</DOJ>
</Employee>
```

Software AG Training | 6 - 31

Notes:

Antivirus Scan Filter

- Enable API Gateway to interact with an Internet Content Adaption Protocol (ICAP) Server
 - to scan all incoming HTTP requests and payload for viruses
- ICAP is capable of hosting multiple services to implement
 - Virus scanning
 - Content filtering
- Pre-requisite
 - ICAP-compliant server installed and configured in DMZ
 - API Gateway must be able to access this server
 - ICAP-compliant server must have an ICAP service registered
 - API Gateway must be able to send emails to that server

Notes:

Workflow: Antivirus Scan Filter

Anti virus scan filter

Enable

ICAP Name: AntiVirusScan ICAP host name or IP address: mcvirusscan.eur.ad.sag

ICAP port number: 8888 ICAP service name: scanServices

The diagram illustrates the flow of traffic through a network security stack. At the top, external clients (represented by a smartphone, laptop, and desktop) connect to an 'Outer Firewall'. Below the firewall is the 'DMZ' (Demilitarized Zone) area, which contains an 'API Gateway Server' and an 'ICAP Server'. An arrow labeled '1' indicates traffic from the clients to the API Gateway Server. An arrow labeled '2' indicates the API Gateway Server sending traffic to the ICAP Server. An arrow labeled '3' indicates the ICAP Server returning results to the API Gateway Server. Below the DMZ is the 'Inner Firewall', which separates the DMZ from the 'Native Server'. An arrow labeled '4' indicates traffic from the API Gateway Server to the Native Server. An arrow labeled '5' indicates traffic from the Native Server back to the API Gateway Server. Finally, an arrow labeled '6' indicates traffic from the API Gateway Server back to the clients.

Software AG Training | 6 - 33

Notes:

JSON / XML Threat Protection Filter

- Block attacks through JSON / XML payload having
 - Infinitely long strings
 - Deeply nested payloads
- Recommended to be combined with Message Size filter
- JSON payload parameter
 - Container depth, object entry count, object entry name length field, array element count, string value length, applicable content type
- XML payload parameter

The screenshot shows a configuration interface for an XML threat protection filter. At the top left, there is a radio button labeled "XML threat protection filter". Below it is a checkbox labeled "Enable". The interface contains several input fields arranged in a grid:

Namespace prefix length	Namespace URI length	Namespace count per element	Child count
Attribute name length	Attribute value length	Attribute count per element	Element name length
Text length	Comment length	Processing instruction target length	Processing instruction data length
Node depth			
Applicable content types	+ Add		

At the bottom right of the interface, there is a footer bar with the text "Software AG Training | 6 - 34".

Notes:

Software AG Training | 6 - 35

Custom Filter

- Invoke service available on API Gateway / Integration Server
 - Custom authentication of external clients in DMZ
 - Logging and auditing in DMZ
 - Custom rules for processing various payloads
 - Extract HTTP headers and payload
 - Act on it as needed by business requirements
 - Add custom headers and custom response codes

Custom filter

Enable

Invoke service

Run as user

Software AG Training | 6 - 35

Notes:

Threat protection dashboards

- The Threat protection dashboards display a variety of charts to provide an overview of threat protection analytics for rules configured in API Gateway
- Threat protection filters - Displays the graphical representation of the events based on the filter violations in the specified time
- Threat protection rules - Displays the graphical representation of the events based on the rule violations in the specified time
- Threat protection events - Displays the threat protection event details like time, filter name, rule name, resource path, server host, and request time

Notes:

The screenshot shows the 'Threat protection dashboards' section of the webMethods Platform. It features three main components:

- Threat protection rules:** A donut chart showing the distribution of rules. The legend indicates two categories: GlobalDoSRule (purple) and GlobalPRule (pink). The chart is mostly pink.
- Threat protection filters:** A donut chart showing the distribution of filters. The legend indicates two categories: PFBase (blue) and GlobalPRule (pink). The chart is mostly blue.
- Threat protection events:** A table listing recent events. The columns are: Time, filterName, ruleName, resourcePath, serverHost, and requestTime.

Time	filterName	ruleName	resourcePath	serverHost	requestTime
2016-09-30 12:53:48.877	DoSFilter	GlobalDoSRule	invoke/pub/date/getCurrentDate	10.60.34.63	2016-09-29 10:00:36 IST
2016-09-30 12:53:48.877	IPFilter	GlobalPRule	invoke/pub/date/getCurrentDate	10.60.34.63	2016-09-29 10:00:36 IST
2016-09-30 12:00:00.000	IPFilter	GlobalDoSRule	invoke/pub/date/getUser	10.60.34.153	2016-09-29 10:00:36 IST
2016-09-30 11:00:00.000	IPFilter	GlobalPRule	invoke/pub/date/getUser	10.60.34.153	2016-09-29 10:00:36 IST

Software AG Training | 6 - 37

Notes:



Exercise 10

- Managing API Threat Protection

Notes:



Global Policies

Notes:

Inefficient One-to-One Policy Definition

The diagram illustrates the inefficiency of one-to-one policy definition. On the left, a developer named Dave is shown with four service icons (S1, S2, S3, S4). On the right, an SOA architect named Andy is shown with four policy icons (P1, P2, P3, P4). Dashed arrows connect each service to its corresponding policy, forming a 1:1 mapping. This represents the manual process of provisioning 70 services with 30 different policies.

70 Services x 30 Policy Definitions

Dave, Developer

Andy, SOA Architect

Software AG Training | 6 - 40

Next, since policies are so crucial in implementing governance, we also took a close look at how policies are managed and provisioned. At the end of the day, policies have to be applied to each service, each schema, each business process etc. But it is not practical to attach policies to assets one by one. Imagine you have 70 services, schemas and other assets. And you have 30 policies. Which policy applies to an asset depends not only on the type of asset but also by other IT and business criteria. If you tried to provision policies one by one, you would have to make 2100 manual policy decisions! And today CentraSite users are already talking about having hundreds of services and assets. Clearly this model will not scale. With CentraSite ActiveSOA, there is a better way...

Software AG Training

A Smarter Way: Global Policy Definition

Security Audit

every "J2EE" where "HR" Criteria

...

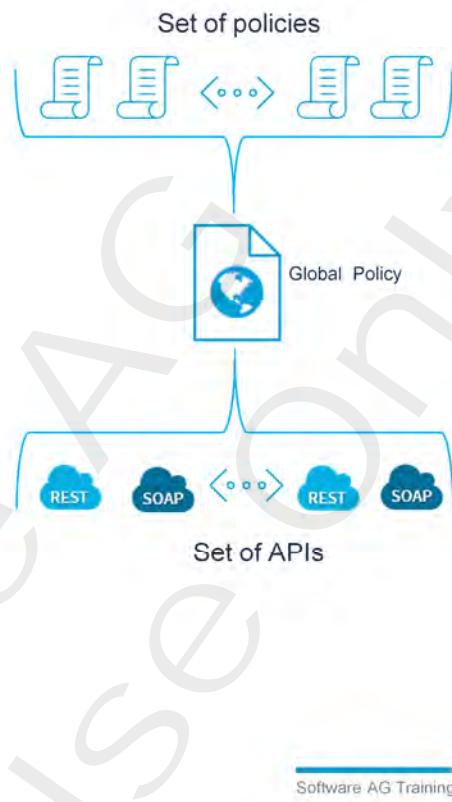
SOA Runtime Governance with effective
Global Policies

Software AG Training | 6 - 41

With CentraSite ActiveSOA you simply define the policy and then provide a criteria for where and when to apply the policy. CentraSite does the rest....

Global Policies in API Gateway

- Associate a group of policy actions to a set of APIs filtered by conditions
- Global policy takes precedence over all other policies (API and scope level policies)
- Modifying a Global Policy will affect all the associated APIs
- Policy modifications can be propagated without downtime

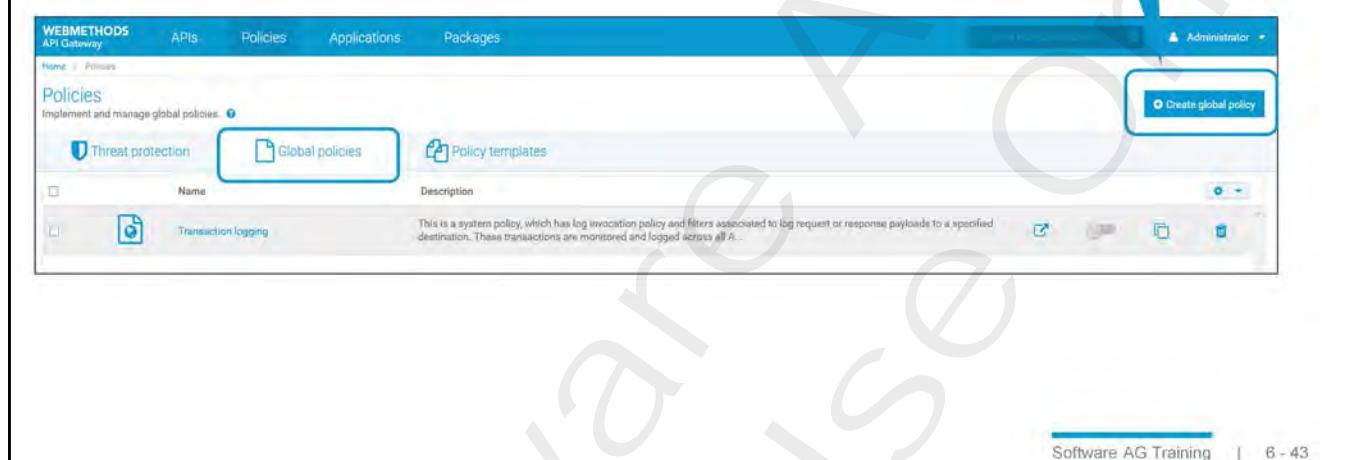


Software AG Training | 6 - 42

Notes:

Global Runtime Policies

- Benefit
 - Dynamic Scope, applicable to multiple Services
- Policies Menu
 - Activity only enabled for user with sufficient permissions
 - API Gateway Administrator



The screenshot shows the 'Policies' section of the WEBMETHODS API Gateway interface. The 'Global policies' tab is selected. A blue callout bubble in the top right corner contains the text 'Add Policy Action'. On the right side of the screen, there is a 'Create global policy' button.

Notes:

Defining a Global Policy – Filter Conditions

- Available Filter Conditions
 - API Type
 - REST / SOAP / OData
 - HTTP Method (for REST only)
 - API Name, API Description, API Version
 - Available operators:
 - A set of Attribute Filter conditions can be defined based on AND / OR operator

Software AG Training | 6 - 44

Notes:

Defining a Global Policy – Policy Actions

- Global policy configuration is API type aware.
 - If filter set to a specific API type (REST/SOAP), only policies pertaining to that type can be configured.
 - If filter set to both REST and SOAP, only common policies appears.

Software AG Training | 6 - 45

Notes:

The screenshot shows the 'Global Runtime Policies - List' page. A specific policy named 'Global Authentication' is selected. The interface includes tabs for 'Policy details' and 'Policy configuration'. The 'Basic information' section shows the policy's name and its status as 'Active'. A blue callout labeled 'Policy Activation' points to the 'Activate' button. Another blue callout labeled 'Actual Search Scope' points to the search bar. A third blue callout labeled 'Actual affected APIs' points to the list of APIs affected by the policy. The 'Filters' section indicates that the policy applies to REST and SOAP APIs using PUT and DELETE methods. The 'APIs' section lists three services: 'Book-store' (REST), 'calculator' (SOAP), and 'EchoOnInternalS' (SOAP). A 'Version' section shows the current version as '1.0'.

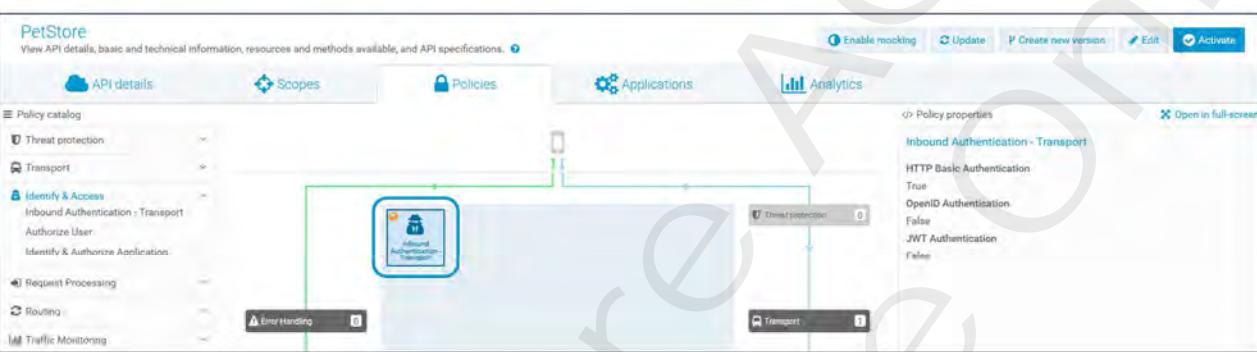
- Policy has 2 lifecycle states
 - Active / inactive
 - Only active Policies will be added to the APIs

Software AG Training | 6 - 46

Notes:

Global Policy Actions in API Policy Definition

- Policy from a Global-policy applied to the API
- Policy with an orange globe is a global policy applicable to a part of the API.
 - Applicable to all Methods except GET



The screenshot shows the Software AG API Management interface for a PetStore API. The top navigation bar includes 'Enable mocking', 'Update', 'Create new version', 'Edit', and 'Activate' buttons. The main area displays a policy catalog with sections like Threat protection, Transport, Identify & Access (highlighted with an orange circle), Request Processing, Routing, and Traffic Monitoring. On the right, a 'Policy properties' panel is open for an 'Inbound Authentication - Transport' policy, showing settings for HTTP Basic Authentication (True), OpenID Authentication (False), and JWT Authentication (False). The bottom right corner of the interface shows 'Software AG Training | 6 - 47'.

Notes:

Deactivating a Global Runtime Policy

The screenshot shows the 'Policies' section of the webMethods Platform. It lists two global policies:

Name	Description
GlobalAuthentication	always enforce HTTP Basic Authentication
Transaction logging	This is a system policy, which has log invocation policy and filters associated to log request or response payloads to a specified destination. These transactions are monitored and logged across all A...

A large blue toggle button is highlighted over the 'GlobalAuthentication' row, indicating it is being deactivated. The 'Lifecycle State' column shows a grey icon for the deactivated policy.

- Switch Lifecycle State of the Policy by toggling the Policy Decorator Indicator (Toggle button)
 - Blue -> active / productive
 - Grey -> inactive

Software AG Training | 6 - 48

Notes:

Changing a Global Runtime Policy

- Global Policies can be modified in any state active/inactive
 - Just switch to Edit mode and invoke the wizard again
 - Modify
 - Filter configuration
 - Policy definitions
 - Save your changes
 - Policy definition in the APIs will be updated to new configuration

Software AG Training | 6 - 49

Notes:

Global policy – Scenario A

- Provide Basic authentication for all APIs in the HR module
- Monitor service performance of all APIs in the HR module

HR Module policy

API name Starts with “HR”

REST HR_EmployeeFinder

REST HR_OrganisationPolicy

Software AG Training | 6 - 50

Notes:

Global policy – Scenario B

- Provide authentication based on API Key for all APIs in the PR module
- Enforce Throttling for all APIs in the PR module

The diagram illustrates the enforcement of a Project Policy for APIs whose names begin with "PR". It features a central "Project Policy" icon containing a globe. Two curly braces extend from this central icon to two separate groups of icons. The left group contains "Identify & Authorize Application" and "Throttling Traffic Optimization" icons. The right group contains a "REST PR_Projects" icon. A downward-pointing arrow labeled "API name Starts with 'PR'" connects the central policy to the right-side icons.

Project Policy

Identify & Authorize Application

Throttling Traffic Optimization

REST PR_Projects

API name Starts with "PR"

Software AG Training | 6 - 51

Notes:



Scope Level Policies

Notes:

Use case for API Scopes

- Shopping API
 - Protected with HTTP Basic Authentication
- Resources
 - /orders**
 - GET
 - POST
 - /orders/{orderId}**
 - GET
 - DELETE
- New Requirement
 - Enforce logging for a specific resource or method
 - Enforce a higher security for POST/DELETE Operation
 - User should be part of a registered consumer application
 -
- Solution
 - Fine granular policy definition for a subset of the API

Software AG Training | 6 - 53

Notes:

API Scopes in API Gateway

- API Scope is a logical grouping of
 - Resources and/or Methods of a REST API
 - Operations of a SOAP API
- Criteria for setting up an API Scope
 - Same set of policy actions that must be enforced for this subset of an API
 - Security / Monitoring
- API Scope rules
 - Each API has its own set of scopes
 - Name of the scope in an API MUST be unique

REST

SOAP

Resources

Operations

GET

POST

PUT

DELETE

PATCH

Scope 1

Scope 2

Scope 3

API 1

Scope 1

Scope 2

API 2

Software AG Training | 6 - 54

Notes:

Defining an API Scope in API Gateway

- API has to be in EDIT mode
- Available for API Gateway Administrator and API Gateway Provider
- Scope has
 - Name and Description
 - The selected group of
 - Operations
 - Resources/Methods
- Defined scopes are visible in
 - Policy Definition Page
 - As a filter to define additional policy actions just for this scope / group / subset of the API



Software AG Training | 6 - 55

Notes:

Policy Definition for API Scopes

- Select the Scope to define a set of policy actions just for members of this group

Software AG Training | 6 - 56

Notes:

Policy Definition for API Scopes

- When switching to a specific scope the Policies defined for the API are visible in grey color - you can not change these
- scope level policies can only be from the sections
 - Identify & Access
 - Traffic Monitoring

Software AG Training | 6 - 57

Notes:

Policy Precedence

- Precedence is determined by the following:
 - Scope level policies for a method/operation
 - Scope level policies for a resource (REST only)
 - API level policies

REST API				SOAP API		
API Level	/res1 (Scope 1)	GET /res1 (Scope 2)	Effective at runtime	API Level	Operation (Scope 1)	Effective at runtime
	N/A	N/A			N/A	
N/A		N/A				
	N/A					
N/A						

Software AG Training | 6 - 58

Notes:

The screenshot shows the Software AG Policy Overview interface. It features three main tabs: API level policies, Scope-level policies, and Resource or method level policies. The Resource or method level policies tab is currently active, displaying a dropdown menu for selecting a resource and method. The dropdown menu lists several options, including '/customer', 'POST - /customer', and '/customer/{customerId}'. The left sidebar contains a navigation tree with categories like Threat protection, Identify & Access, Transport, Request Processing, Traffic Monitoring, Routing, Response, and Error Handling. The top right corner of the interface has a watermark reading "Software AG Internal Use Only".

Notes:



Exercise 11

- Managing Fine Granular Exposure of APIs

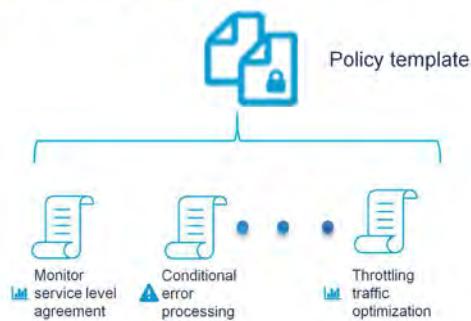
Notes:



Policy Templates

Notes:

Use Case for Policy Templates



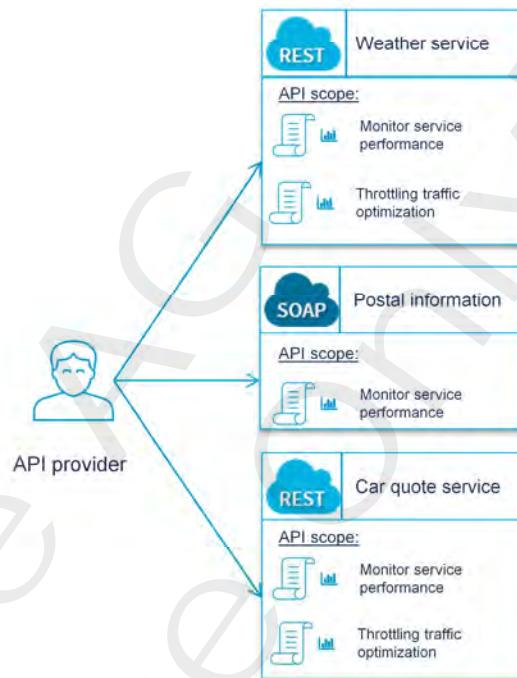
- **Policy Template**
 - a user-defined, high level policy => blueprint of a policy
 - Group of policy actions which work well together
- **Usage**
 - this policy template can be imported into any number of API's
- **Benefit**
 - Make use of an already defined and tested set of policy action for more than just 1 API

Software AG Training | 6 - 62

Notes:

Policy Template Use Case

- API Provider would like to have
 - in all of his APIs
 - Monitor Service Performance
 - In 2 of his APIs
 - Monitor Service Performance
 - Throttling
 - Traditional configuration
 - Create global Policy
 - Which will not help for the Postal service
 - Define the policies for each API
 - Definition of 5 policy actions



Software AG Training | 6 - 63

Notes:

Policy Template Definition from Scratch

▪ Define a policy template

- Including Monitoring and Throttling

 API provider → **Traffic monitoring template**

- Monitor service performance
- Throttling traffic optimization

▪ Apply the policy template to all 3 APIs

▪ Remove the Throttling policy in the Postal service

```

graph TD
    TM[Traffic monitoring template] --> WS[Weather service]
    TM --> PI[Postal information]
    TM --> CQS[Car quote service]
    subgraph WS [Weather service]
        REST_WS[REST]
        PI_WS[Postal information]
        APISCOPE_WS["API scope:"]
        APISCOPE_WS --> R_WS[Require HTTP/HTTPS]
        APISCOPE_WS --> SR_WS[Straight through routing]
        APISCOPE_WS --> MPS_WS[Monitor service performance]
        APISCOPE_WS --> TTO_WS[Throttling traffic optimization]
    end
    subgraph PI [Postal information]
        REST_PI[REST]
        PI_PI[Postal information]
        APISCOPE_PI["API scope:"]
        APISCOPE_PI --> R_PI[Require HTTP/HTTPS]
        APISCOPE_PI --> SR_PI[Straight through routing]
        APISCOPE_PI --> MPS_PI[Monitor service performance]
        APISCOPE_PI --> TTO_PI[Throttling traffic optimization]
        TTO_PI -.-> TTO_PI
    end
    subgraph CQS [Car quote service]
        REST_CQS[REST]
        PI_CQS[Postal information]
        APISCOPE_CQS["API scope:"]
        APISCOPE_CQS --> R_CQS[Require HTTP/HTTPS]
        APISCOPE_CQS --> SR_CQS[Straight through routing]
        APISCOPE_CQS --> MPS_CQS[Monitor service performance]
        APISCOPE_CQS --> TTO_CQS[Throttling traffic optimization]
    end

```

Software AG Training | 6 - 64

Notes:

Policy Template Definition from Existing API Policy Definitions

- Define and save all policy actions for 1 API
- Use the Save as template feature to export the policies in API scope as a new policy template
- Delete the policy actions not needed in the policy template
- Apply the policy template to the other 2 APIs

The diagram illustrates the workflow for creating a policy template. On the left, an 'API provider' is shown interacting with a 'Weather service' interface. The 'Weather service' interface displays its 'API scope' with four policy actions: 'Require HTTP/HTTPS', 'Straight through routing', 'Monitor service performance', and 'Throttling traffic optimization'. A blue arrow points from the 'API provider' to the 'Weather service'. From the 'Weather service' interface, a blue arrow points to a 'Save as template' button. This button is highlighted with a red border, indicating it is being used. A second blue arrow points from the 'Save as template' button to a 'Traffic monitoring template' interface on the right. The 'Traffic monitoring template' interface shows the same four policy actions, but the first two ('Require HTTP/HTTPS' and 'Straight through routing') are crossed out with a large red 'X'. Below these crossed-out policies, the text 'Delete' is written in red. The 'Traffic monitoring template' also includes two additional policy actions: 'Monitor service performance' and 'Throttling traffic optimization'.

Notes:

The screenshot shows the Policy Template Management interface for the 'AnnesFirstTemplate'. The top navigation bar includes links for API details, Scopes, Policies, Applications, and Analytics. The main title 'Policy Template Management' is displayed above the template configuration area.

The left sidebar contains a 'Policy catalog' with categories: Transport (Require HTTP / HTTPS), Identify & Access, Request Processing, Routing, Traffic Monitoring, Response Processing, and Error Handling. The 'Transport' category is currently selected, showing its sub-options.

The central workspace displays the 'AnnesFirstTemplate' configuration. It features a flowchart with nodes: 'Error Hand.', 'Response...', 'Identity...', 'Transport...', 'Routing...', 'Request...', 'Response...', and 'Error Hand.'. Arrows indicate the flow between these nodes. A green box highlights the 'Routing...' node. The right side of the screen shows 'Policy properties' such as 'Protocol: HTTP' and 'Require HTTP / HTTPS'. There are also 'Edit' and 'Cancel' buttons.

At the bottom right of the interface, the text 'Software AG Training | 6 - 66' is visible.

Notes:

The screenshot shows the Bookstore application's 'Apply Template' dialog. On the left, a sidebar lists policy categories: Threat protection, Transport, Identify & Access, Request Processing, Routing, Traffic Monitoring (selected), Response Processing, and Error Handling. The main area displays a table of templates:

Name	Description
AnnotFirstTemplate	

Below the table, a detailed view of the 'AnnotFirstTemplate' shows its application across various API components:

- Transport:** 1 Policy(s) applied: RequireHTTP / HTTPS.
- Identify & Access:** 2 Policy(s) applied: Identify & Access Authorization, Inbound Authentication - Transport.
- Request Processing:** 0 Policy(s) applied.
- Traffic Monitoring:** 2 Policy(s) applied: Throttling Traffic Optimization, Log Invocation.
- Routing:** 1 Policy(s) applied: Straight Through Routing.

A note at the bottom right states: "After applying specific templates to the API, the policies associated with the template are associated to the API. You can modify and remove a policy from the API."

Software AG Training | 6 - 67

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



7

Traffic Management

Notes:

Objectives

At the end of this chapter you ...

- Know how to manage the overall performance of your APIs
- Know how to enforce SLAs

Notes:

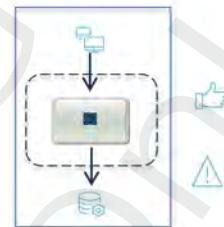
Traffic Management

▪ Logging & Monitoring

- Monitor and collect information about the number of messages processed (success or failure, number of calls, average/min/max response time)
 - Log Invocation
 - Monitor Service Performance
 - Monitor Service Level Agreement

• Traffic Management

- Avoid overloading the back-end services
 - Throttling Traffic Optimization (limit the number of calls)
 - Service Result Cache



Software AG Training | 7 - 3

Notes:

API Consumer Management

- Logging of external access
- Monitoring of a service by configuring rules that are based on service performance parameters
 - Collect runtime data
 - Aggregate runtime data
 - Alert on wrong usage
- Service Result Caching
 - Reduce latency
- Monitoring of a service based on consumers
 - Different rules for different classes of consumers (Gold / Silver customer)
- Traffic Optimization – regulate the asset usage
 - Protect your backend services by constraining the traffic
 - Support different levels for different customers
 - Improve throughput and latency

Notes:

Event Configuration – Globally in Gateway Administration

- Controlled via API Gateway
- Events are per invocation
- Events Types which are always turned ON / always sent:
 - Transaction Events (example: logging of transactions)
- Event Types can be turned on:
 - Policy Violation (example: Violation of a security policy)
 - Error Events (example: backend service unavailable)
 - Monitoring Events (example: SLA monitoring)
- Possible destinations for Events
 - API Gateway (default setting)
 - API Portal
 - CentraSite
 - Database
 - Digital Events
 - Elastic Search
 - API-Portal
 - Email
 - SNMP

Notes:

The screenshot shows the Software AG Analytics interface. At the top, there are tabs for 'API details', 'Policies', 'Applications', and 'Analytics'. The 'Analytics' tab is selected. Below the tabs, there are filters for 'Last 12 Hours', 'From Date', 'To Date', and 'Apply Filter'. A bar chart titled 'Events per hour' shows event counts for different time intervals. A legend indicates three categories: PerformanceData (purple), PolicyViolation (red), and Transactional (orange). To the right, a circular chart titled 'Event types' shows the distribution of event types. Below these charts, a table lists 'Selected Events' with columns for Time, opName, applicationName, and eventType. The table entries are as follows:

Time	opName	applicationName	eventType
2017-02-16 12:25:20.391	SearchCruise	Unknown	PolicyViolation
2017-02-16 12:37:07.168	SearchCruise	-	PerformanceData
2017-02-16 13:23:54.281	SearchCruise	Unknown	Transactional
2017-02-16 12:59:07.212	SearchCruise	-	PerformanceData
2017-02-16 13:31:07.217	SearchCruise	-	PerformanceData

Software AG Training | 7 - 6

Notes:

Event Types in API Gateway

- Overall Events

Event Type	Description
Lifecycle	Occurs each time API Gateway is started or shut down
Error	Occurs each time an invocation of a virtual service results in an error
Policy Violation	Occurs each time an invocation of a service violates a run-time policy
Transactional	API Gateway publishes events in case a monitoring run-time policy action is configured (Logging, Monitoring)
Performance Data	Occurs based on the performance reporting interval

Notes:

Software AG Training | 7 - 8

Log Invocation

- Log Invocation Policy is used to record the invocations of an API.
- The policy generates a **Transactional** event for every API request.
- The policy has provisions to configure
 - Log generation frequency
 - Destination for the generated events
 - Payload storage and its format

The screenshot shows a configuration dialog for 'Log Invocation'. It includes sections for 'Store Request Payload' (checked), 'Store Response Payload' (checked), 'Compress Payload Data' (unchecked), 'Log Generation Frequency' (set to 'Always'), and a 'Destination' section. In the 'Destination' section, 'API Gateway' and 'Elasticsearch' are checked, while other options like 'Audit Log', 'CentraSite', 'Digital Events', 'Email', 'JDBC', 'Local Log', and 'SNMP' are unchecked.

Software AG Training | 7 - 8

Notes:

Monitor Service Performance

- Allows a user to monitor a service by configuring rules that are based on service performance parameters.
 - Action Configuration
 - Destination for the event generated
 - Alert interval and unit
 - Alert Frequency
- Allows a user to identify if a service level threshold has or has not been met.
- User can add multiple monitoring policies actions for a particular service.
- If one of the rules is violated, monitoring alert events are generated.

Monitor Service Performance

Action Configuration *

Add action configuration

Name

Total Request Count

Availability

Average Response Time

Fault Count

Maximum Response Time

Minimum Response Time

Success Count

Total Request Count

Cancel Add

Destination *

API Gateway

API Portal

CentraSite

Digital Events

Elasticsearch

Email

JDBC

Local Log

SNMP

Alert Interval

1

Unit

Notes:

Monitor Service Level Agreement

- The policy can be calculated for a particular consumer.
 - An Identify & Authenticate Consumer Policy Action must be configured to determine the assignment of the request to a consumer application
 - API Gateway will add this policy if not already added
 - KPIs are evaluated at the end of the alert interval (1-60 min) (For metrics such as avg/min/max response time. For other metrics such as fault / success / total count the rule is evaluated for each service request).
 - Alert Frequency can be set to generate alerts “Every time” or “Only Once” on rule violation.
 - Alert Destination
 - Alert message(s) can be defined.

Monitor Service Level Agreement

Destination *

- API Gateway
- API Portal
- CentraSite
- Digital Events
- Elasticsearch
- Email
- JDBC
- Local Log
- SNMP

Alert Interval

5

Unit

Minutes

Alert Frequency

Only Once Every Time

Alert Message*

Limit of Average response time exceeded

Consumer Applications *

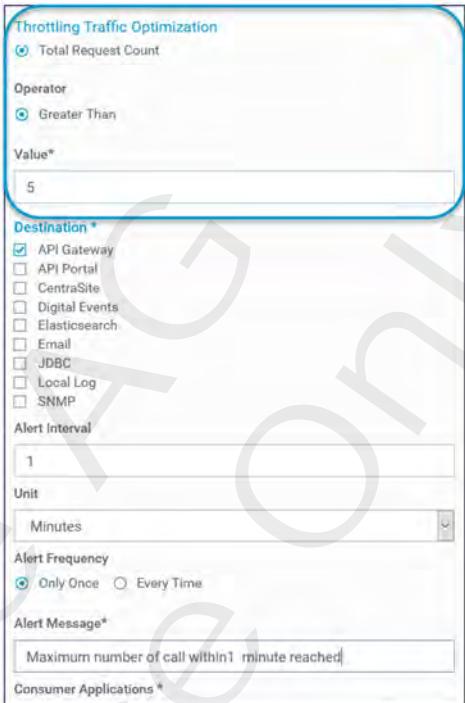
a.	x
a shopping application	x
Bookstore App	x
Customer Application	x

Software AG Training | 7 - 10

Notes:

Runtime Traffic Management

- Runtime Traffic Management allows for predictable operations
 - Amount of traffic reaching backends can be controlled
 - Avoid overloading of slow/loaded backends
- Enforce consumption SLA's for your service consumers
 - Privileged consumers vs. Normal consumers based on Consumer Applications
- The policy generates a Policy violation event when the incoming request for the API is throttled

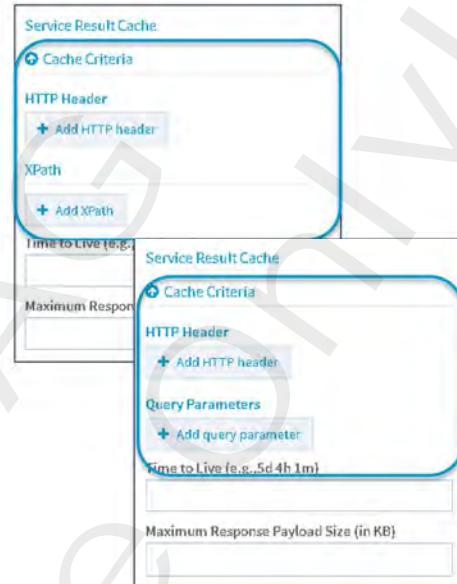


Software AG Training | 7 - 11

Notes:

Service Result Caching

- Cache service responses of the native service to improve
 - Throughput and latency
 - Use cases:
 - Faster response
 - Reduce network calls
 - Helps when server is down temporarily
 - Instead of invoking the native API for every client call, providers can configure caching
 - NOT to use when consumers
 - Absolutely rely on current information retrieved from back-end service
 - Configuration options
 - Cache key criteria
 - Only 1 criteria is supported
 - Time-to-Live for Cache Data
 - Maximum Response Payload Size
 - Applicable to SOAP and REST



Software AG Training | 7 - 12

Notes:

Configuring Service Result Cache

- Caching criteria

- Determines the request component that is the actual payload based on which the result of the API invocation is cached
 - HTTP Header – for REST and SOAP based API that accept payloads only in HTTP format
 - Query Parameter – Query parameters names, mainly for REST APIs
 - XPath – XPath expression, mainly for SOAP based request (payload is a SOAP Envelope)
- Add to Whitelist (optional)
 - Criteria evaluation (based on caching criteria) must result in any of the values in this list before API Gateway caches the API response

- Time to Live (optional)

- Days, hours, minutes after which elements in the cache are considered to be outdated

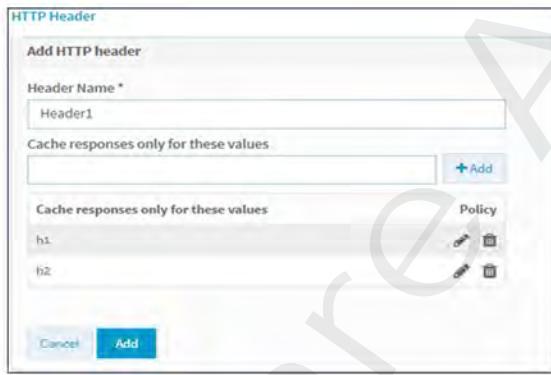
- Maximum Response Payload

- Maximum payload size in kb
- -1 defined an unlimited payload size

Notes:

Service Result Caching - Limitations

- Configure cache criteria properly that defines the payload
 - What is the KEY for the lookup in the cache => unique identifier
 - Make sure to identify the correct data – you need to know the data context/content
- Caching results will vary depending on your backend service responsiveness
 - Back-ends with low latency / response times will not greatly benefit



Software AG Training | 7 - 14

Mediator uses Ehcache capability provided by Integration Server to cache the results of the API calls. You can configure caching for a single Mediator node or for a cluster.

Recommendations and Best practices:

- Caching will occur in memory, ensure that you are not operating in a memory constrained environment
- Balance the memory consumption with the API response sizes. If your API returns huge responses or large binary attachments, limit caching by specifying the maximum size of the cache or by defining data eviction policies to avoid excessive consumption
- Design APIs for which you want to implement caching to be idempotent to support reliable caching
- Configure a suitable value for the Time to Live parameter for the cache entries based on your business needs and the use case for your API
 - If your API serves static product catalog data which is only updated once a quarter choose a large TTL value. If your API serves for example environmental data where a certain age of data is tolerated, choose a TTL value like 15 minutes

Caching recommended

data does not change frequently, request to DB has a high response time
 client can use slightly outdated, cached data – weather API
 temporary service interruptions

No caching:

response or DB access is very fast, API uses non-idempotent requests



Exercise 12

- Monitoring the Virtual API

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



8

Routing and Mediation

Notes:

Objectives

At the end of this chapter you ...

- Know how to route to a specific native endpoint
- How to do transformation of request / response / error message

Notes:

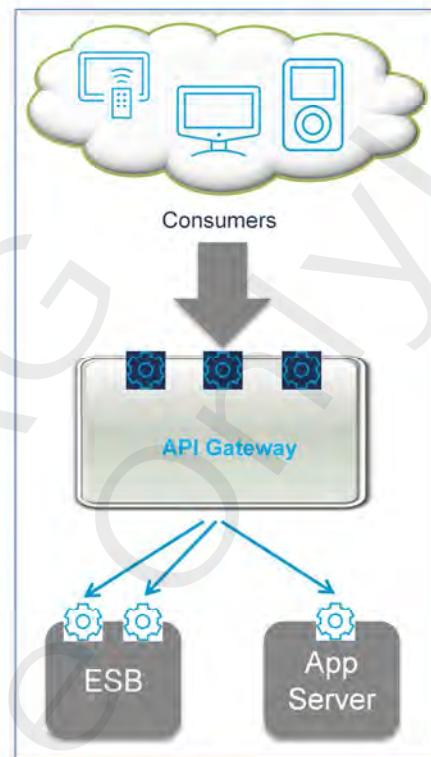
Chapter Contents

- Message Transformation
- Error Processing
- Protocol Transformation
- SOAP to REST Transformation

Notes:

Mediation Use-Cases

- Route the order request to the nearest fulfillment provider based on location passed in the Protocol header.
- Route request to backend services located in another domain
- Comply with backend service security policies by adding Security tokens on the fly while forwarding requests.
- Invoke backend .NET services using WS-Addressing



Software AG Training | 8 - 4

Notes:

Message Processing In API Gateway

Policy catalog

- Threat protection
- Transport
- Identity & Access
- Request Processing
 - Routing** (selected)
 - Content-based Routing
 - Context-based Routing
 - Dynamic Routing
 - Load Balancer Routing
 - Straight Through Routing
 - Custom HTTP Header
 - Outbound Authentication - Transport
- Traffic Monitoring
- Response Processing
- Error Handling

Policy properties

Straight Through Routing

Endpoint URI: http://localhost:5555/re/bookstore

Routing Method: CUSTOM

HTTP Connection Timeout (seconds): 30

Read Timeout (seconds): 30

SSL Configuration

Keystore Alias: Enter keystore alias to use SSL/TLS

Key Alias: Enter key alias to use SSL/TLS

Global Policy | Scope | Conflict | Routing

Software AG Training | 8 - 5

- Routing Policies
 - Content-based Routing
 - Context-based Routing
 - Dynamic Routing
 - Load Balancer Routing
 - Straight Through Routing
 - Custom HTTP Header
 - Outbound Authentication Transport

Notes:

Routing Processing

- Configure Endpoint Routing
 - Straight Through Routing
 - Predefined
 - Points to the defined native endpoint
 - Content based
 - Introspect message through XPath expression
 - Context based
 - Routing decisions through context information like consumer, operation, date/time,...
 - Load balanced Routing
 - Automatic Endpoint Failover/Retry
 - Dynamic Routing
 - Routing decision based on HTTP header or context information
- Configure HTTP Header
- Configure Outbound Authentication Transport

The diagram illustrates the routing process. At the top is a box labeled 'API Gateway' with three gear icons. A large downward-pointing arrow originates from the gateway. From the middle of this arrow, two smaller arrows branch out to two separate boxes below: 'IS' (with two gear icons) and 'App Server' (with one gear icon). This visualizes how the API Gateway directs traffic to different internal service components.

Software AG Training | 8 - 6

Notes:

Configure Routing Steps

Straight Through Routing

Endpoint URI * http://DAETRAIN00749.eur.ad.sag:5555/ws/SAGTours.WEBServices.Yach

Read Timeout (seconds) 30

SSL Configuration

SOAP Optimization Method MTOM SwA None

HTTP Connection Timeout (seconds) 30

Pass WS-Security Headers

Straight Through Routing

Endpoint URI * http://localhost:53307/SAGTours/\$[sys/resource_path]

HTTP Connection Timeout (seconds) 30

SSL Configuration

Keystore Alias Enter search terms to see suggestions

Key Alias Enter search terms to see suggestions

Routing Method GET POST PUT DELETE CUSTOM

Read Timeout (seconds) 30

Software AG Training | 8 - 7

Notes:

The diagram illustrates a consumer application sending a request to an API Gateway. The API Gateway then routes the request to either a default endpoint or a specific endpoint for the 'multiply' operation.

```

graph LR
    CA[Consumer Application] -- "application/xml  
Calculate operation  
\"add\", \"multiply\", ..." --> AG[API Gateway]
    AG -- "default" --> DS[CalculatorService]
    AG -- "multiply" --> D[Call \"multiply\" operation on server \"Dummy\"]
    DS -- "Call operation on server \"localhost\"" --> DS
  
```

Content-based Routing Configuration:

- Default Route To:**
 - Endpoint URI: `https://localhost:5555/wa/calculator/calculator_calculate`
 - Read Timeout (seconds): 30
- Rules:**
 - Rules** (selected)
 - XPath Expression**: `/soapenv:Envelope/soapenv:Body/calmultiply_flowservice`
 - Namespace**: `http://www.soapenv.org/`
 - Route To:**
 - Endpoint URI: `tcp://Dummy1:5555/wa/calculator/calculator_calculate`
 - HTTP Connection Timeout (seconds): 30
 - Read Timeout (seconds): 30

Software AG Training | 8 - 8

Notes:

The diagram illustrates a context-based routing scenario. An 'Order Consumer' sends an 'application/xml' request to an 'API Gateway Context Variable "User"'. The 'API Gateway' then routes the request to either the 'USFulfillmentFlowService' (which uses FEDEX) or the 'EUFulfillmentFlowService' (which uses DHL).

Context-based Routing Configuration:

- Default Route To:**
 - Endpoint URI:** http://localhost:5555/wa/Shipping:USFulfillmentFlowService_WSDL/
 - Read Timeout (seconds):** 30
 - SSL Configuration:**
 - SOAP Optimization Method:** MTOM
- Rules:**
 - Name:** UserLocationShipping
 - Condition Operator:** Or
 - Condition:**
 - Condition:** Predefined Context Variable
 - Variable:** User
 - Value:** John
 - Operator:** Equal To
 - Route To:**
 - Endpoint URI:** http://localhost:5555/wa/Shipping:EUFulfillmentFlowService_WSDL/
 - Read Timeout (seconds):** 30
 - SSL Configuration:**

Software AG Training | 8 - 9

Notes:

The screenshot shows a configuration interface for 'Predefined Context Variables'. On the left, a 'Condition' panel has a 'Variable *' dropdown containing 'Consumer', 'Date', 'IPv4', 'IPv6', and 'Predefined Context Variable'. The 'Predefined Context Variable' option is selected and highlighted with a blue border. A blue arrow points from this selection to the right-hand panel. The right-hand panel also has a 'Variable *' dropdown labeled 'Predefined Context Variable', which is currently set to 'User'. Below this are several other options: 'Inbound HTTP Method', 'Routing Method', 'Inbound Content type', 'Inbound Accept', 'Inbound Protocol', 'Inbound Request URI', 'Inbound IP', 'Gateway Hostname', 'Gateway IP', 'SOAP Header', 'Protocol Header', and 'Operation Name'. To the right of the dropdown is an 'Operator' section with a radio button for 'Equal To' selected.

- Definition of values to match application
 - range for
 - Date
 - Time
 - IPV4
 - IPV6
 - Variable value for
 - Consumer
 - Predefined Context Variable

Software AG Training | 8 - 10

Notes:

The diagram illustrates the architecture for routing requests from an Order Consumer to an OrderService. An Order Consumer sends a request to an API Gateway with a custom variable (application/xml). The API Gateway then routes the request to two different endpoints based on custom variables defined in the configuration.

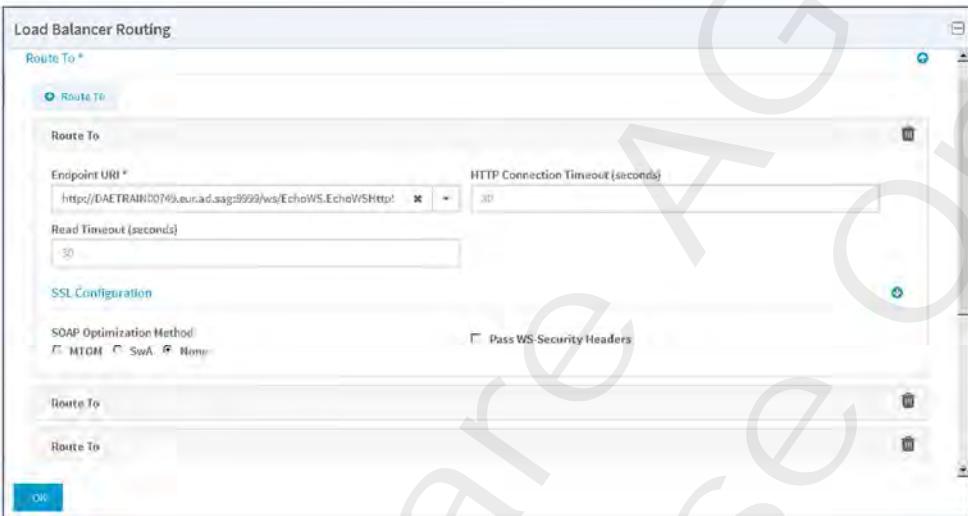
Dynamic Routing Configuration:

- Default Route To:**
 - Endpoint URI: `AIN00749,eur.ad.sag:9999/wa/EchoWS.EchoWSHttpSoap12Endpoint`
 - HTTP Connection Timeout (seconds): 30
- Rule:**
 - Route Using: Header Context
 - Name: demo
 - Route To:
 - Endpoint URI: `http://DAETRAIN00749.eur.ad.sag:9999/wa/Sysdyn-Endpoint`
 - HTTP Connection Timeout (seconds): 30

Notes:

Load Balancer Routing

- Load Balancer Routing Definition
 - A number of Endpoint URIs
- Load Balancing always works based on
 - Round Robin



Software AG Training | 8 - 12

Notes:

Outbound Authentication - Transport in API Gateway

The diagram illustrates the flow of data through an API Gateway. It starts with an 'Error Handling' step, followed by 'Outbound Authentication - Transport' (with a blue icon), then 'Through Routing' (with a green icon), 'Threat protection' (with a grey icon), and finally 'Transport' (with a blue icon).

Outbound Authentication - Transport

Authentication scheme

- Basic
- Kerberos
- NTLM
- OAuth2
- JWT
- Anonymous
- Alias

Username:

Software AG Training | 8 - 13

Notes:

Message Processing for SOAP APIs



- Verify that the client has the proper credentials to access the API
- Authentication Scheme: specifies the mode of authentication API Gateway use to invoke the native service
 - None
 - WSS Username
 - Kerberos
 - SAML
 - Provide SAML issuer that is configured
 - Alias
- Signing and Encryption Configurations
 - Provide Keystore, Truststore and Certificate information
- Authentication definition can co-exist with Outbound Auth. Transport

Outbound Authentication - Message

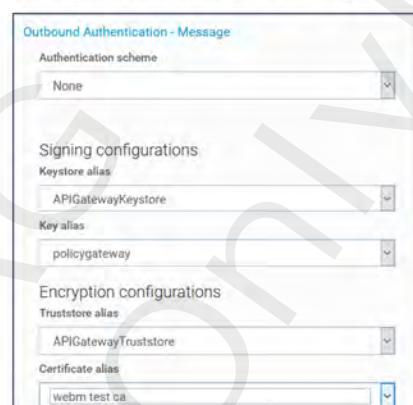
Authentication scheme	None
Signing configurations	
Keystore alias	APIGatewayKeystore
Key alias	policygateway
Encryption configurations	
Truststore alias	APIGatewayTruststore
Certificate alias	webm test ca

Notes:

Outbound Authentication - Message in API Gateway

- Verify that the client has the proper credentials to access the API
- Authentication Scheme: specifies the mode of authentication API Gateway use to invoke the native service
 - None
 - WSS Username
 - Kerberos
 - SAML
 - Provide SAML issuer that is configured
 - Alias
- Signing and Encryption Configurations
 - Provide Keystore, Truststore and Certificate information
- Authentication definition can co-exist with Outbound Auth. Transport



Software AG Training | 8 - 15

Notes:



Exercise 13

- Using a Routing Policy Action

Notes:



Message Transformation

Notes:

Message Transformation

- Flexibility in message formats and transports for consumers
 - Message Formats
 - Error Handling
- Use Cases:
 - Native Service can change request message independent from client application
 - Client application can send request message as needed – with or without specific header information
 - Client application expects custom specific error handling
 - API Gateway will transform the messages and enforce client specific error handling

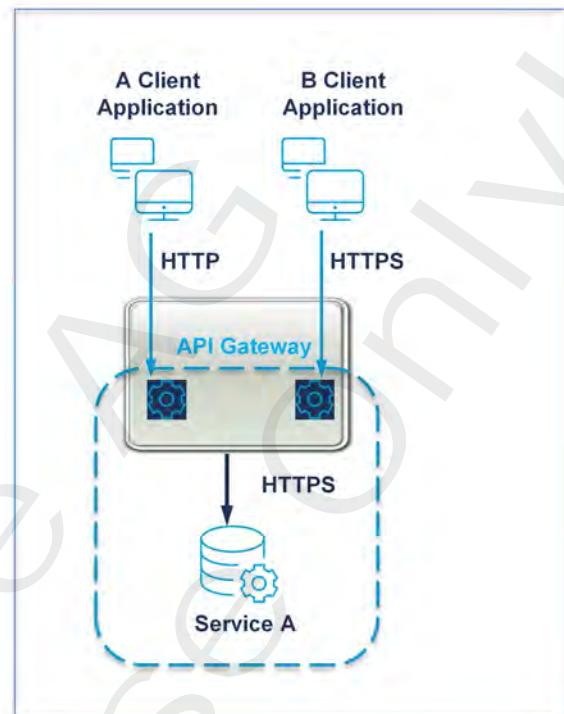


Software AG Training | 8 - 18

Notes:

Request Handling

- Expect Transport Format
 - Require Protocol
 - Require HTTP / HTTPS
 - Require JMS
 - Validate HTTP Headers
 - Set Media Type (for REST APIs)
- Transform request message to ensure compatibility
 - modify a request, add or remove headers in the request, add security information etc.
 - Request Message Transformation
 - XSLT Transformation
 - Invoke webMethods IS
 - Validate Schema

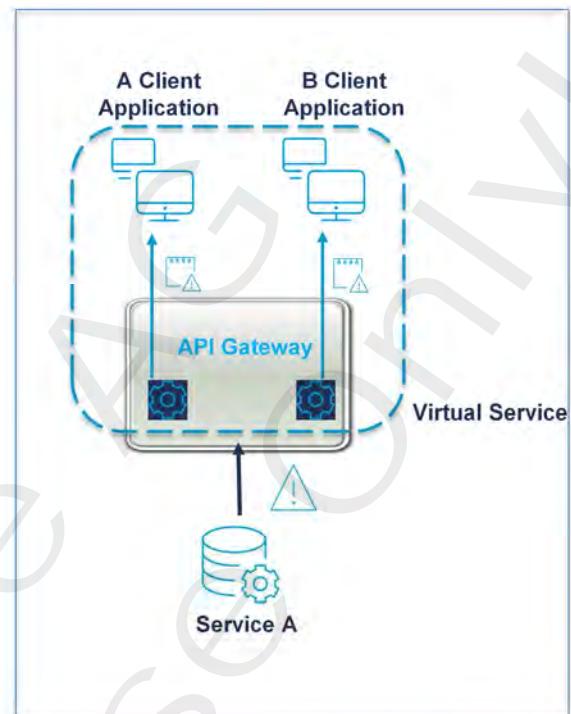


Software AG Training | 8 - 19

Notes:

Response Processing and Error Handling

- Transform response message to ensure compatibility
 - Request Transformation
 - XSLT Transformation e
 - Invoke webMethods IS
 - Validate Schema
- Error Handling
 - Error Conditions
 - Failure Message
 - Custom Error Variables
 - Pre- and Post-Processing



Software AG Training | 8 - 20

Notes:

Message Transformation

- Transport
- Request Response Processing
- Response Processing
- Error Handling

Software AG Training | 8 - 21

Notes:

Request and Response Processing

- Invoke webMethods IS
 - API Gateway sends MessageContext as an input to the service during the invoke.
 - While writing the “webMethods IS Service”, users can obtain the request/response from the input MessageContext and transform the request/response as per their needs.
- XSLT Transformation
 - API Gateway reads the XSL instructions in the XSL file attached and transforms the requests/responses as per the instructions

The screenshot shows two configuration panels side-by-side. On the left is the 'Invoke webMethods IS' panel, which has a single input field 'Test:ConcatInputJavaSvd' and an 'OK' button. On the right is the 'XSLT Transformation' panel, which includes fields for 'XSL Document', 'XSLT Document', 'XSLT File' (containing 'XSLT_Transformation'), and 'XSLT Features'. Both panels have an 'OK' button at the bottom.

Software AG Training | 8 - 22

Notes:



Error Processing

Notes:

Error handling in API Gateway

- Error handling can cover
 - Error/Exceptions sent by the native provider
 - Errors generated from API-Gateway
- Use Cases:
 - Unique handling for different error scenarios
 - Configure error message templates
 - Transformation of the error message send by the Native provider by using XSLT transformations
 - Processing the error message send by the Native provider using webMethods ESB services

The screenshot shows the Software AG Policy catalog interface. On the left, there is a navigation tree with categories like Threat protection, Transport, Identify & Access, Request Processing, Routing, Traffic Monitoring, Response Processing, and Error Handling. Under Error Handling, 'Conditional Error Processing' is selected and highlighted with a green box. In the main workspace, a flowchart is displayed. It starts with a 'Request' icon, followed by a 'Conditional Error Processing' node (also highlighted with a green box). This is followed by another 'Conditional Error Processing' node, which then leads to a 'Transport' icon. The flowchart is enclosed in a large rounded rectangle.

Software AG Training | 8 - 24

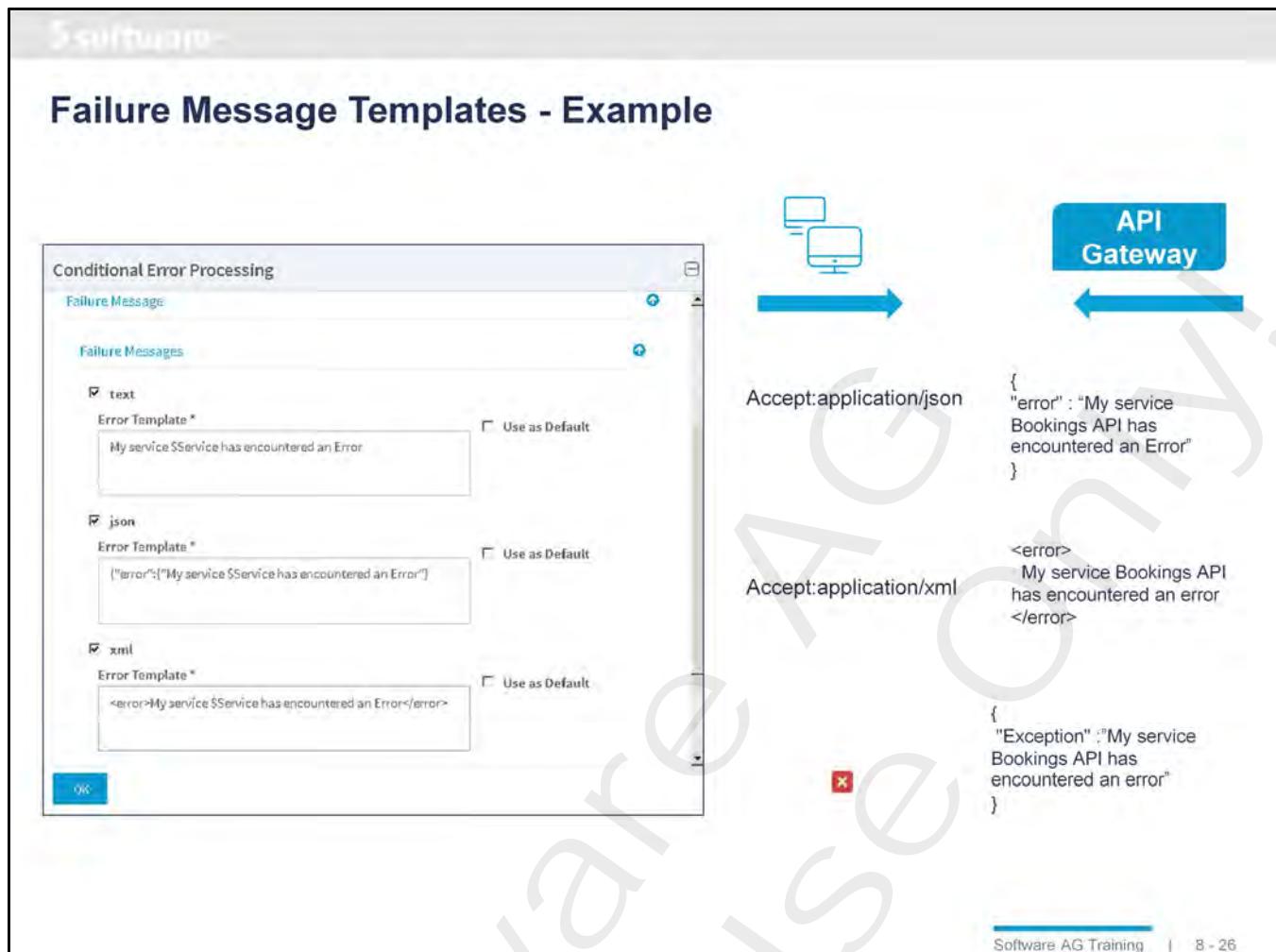
Notes:

Error processing capabilities

- Error Conditions
 - Execute an error step based on an AND combination of error criteria like
 - Status Code: 4xx and 5xx
 - HTTP Header added by native endpoint
 - XPath expression
- Failure Message
 - Extensibility to configure failure messages as templates
 - HTML templates
 - JSON templates
 - Text Message templates (Default Template)
- Pre- and Post-Processing
 - This is an Extension point where provider can run an XSLT transformation or an ESB service during the error processing step. This can be done at two points
 - Before and After API Gateway process the error
- Custom Error Variables



Notes:



Lets see the usage of Failure message templates with an example.
Consider the API-Provider has following configuration for his API.

With respect to this configuration, lets see the runtime usage.
If the Client send an Request with Accept : 'application/json'
that has caused an error as per the configuration the API Gateway
returns the error message.

This is the same as for Accept :application/xml.

If an Accept header is not present in the request then API Gateway uses the text
message and
wraps it with the 'Exception' tag as it is marked as default in the configuration.



Exercise 14

- Updating Content Using XSLT Stylesteet

Notes:



Protocol Transformation

Notes:

Java Messaging Service (JMS) Support in API Gateway

- Use cases
 - JMS -> HTTP (reliability, scalability, asynchronous messaging)

```
graph LR; Clients[Clients] -- JMS --> APIGateway[API Gateway]; APIGateway -- JMS --> API[API]; API -- HTTP --> MissionCriticalSystem[Mission-critical system<br/>SOAP services]
```

- HTTP -> JMS (Interoperability)

```
graph LR; ExternalSystem[External system] -- HTTP --> APIGateway[API Gateway]; APIGateway -- JMS --> Intranet[Intranet]; Intranet -- JMS --> JMSComponents[JMS components]
```

Software AG Training | 8 - 29

- Java Messaging Service
 - Loosely coupled, reliable, asynchronous...
 - Point-to-point
 - Publish-subscribe

JMS Inbound – Require JMS



- Native service accepts requests through HTTP protocol, whereby the consumer send a request via JMS
- API on API Gateway for native API, exposed over HTTP
 - accepts requests sent from a JMS provider queue
 - Listening from JMS provider queue for incoming requests
 - Send the response to another JMS provider queue
 - HTTP native service is exposed over JMS through API Gateway using protocol bridging
- JMS Inbound policy: Require JMS



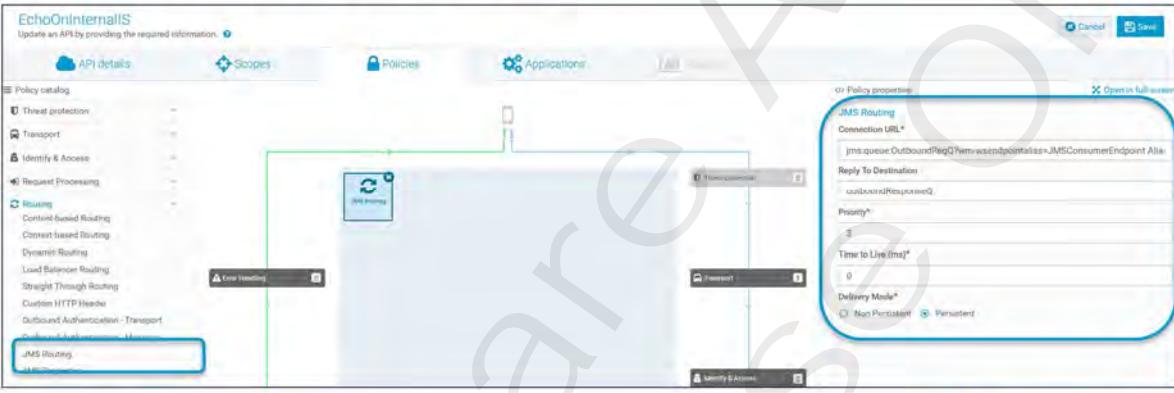
Software AG Training | 8 - 30

Notes:

JMS Outbound – JMS Routing



- Native service accepts requests using JMS protocol, whereby the consumer sends a request over HTTP protocol
- API on API Gateway for native API, exposed over JMS
 - accepts requests sent from a HTTP client
 - Listening from HTTP clients for incoming requests
 - Send the response to back to HTTP client through API Gateway protocol bridging
- JMS Outbound policy: JMS Routing



The screenshot shows the 'EchoOnInternalIS' API configuration screen. The 'Routing' section is selected in the left sidebar. A 'JMS Routing' policy is highlighted with a blue box. On the right, a detailed configuration dialog for the 'JMS Routing' policy is open, also with a blue box around its content. The dialog contains fields for 'Connection URL', 'Reply To Destination', 'Priority', 'Time to Live (ms)', and 'Delivery Mode' (with 'Non Persistent' and 'Persistent' options).

Notes:

The screenshot shows the 'JMS Outbound – JMS Properties' configuration screen. On the left, there's a sidebar with categories like Threat detection, Transport, Identify & Access, Request Processing, and Routing. Under Routing, 'JMS Properties' is selected and highlighted with a red box. The main workspace shows a flow diagram with nodes: 'JMS Routing' (with a red box), 'JMS Properties' (highlighted with a red box), 'Error Handling', 'Timeout', 'Transport', and 'Identity & Access'. A modal window titled 'JMS Properties' is open, showing a table of properties:

JMS Property Key*	JMS Property Value*	Action
jms.timeToLive	30000	
SOAPJMS_targetService	Another API Aggregator API	
JMS Type	ABC	

Below the table, there are buttons for 'Cancel' and 'Save'. At the bottom right of the slide, there's a footer bar with 'Software AG Training | 8 - 32'.

Notes:

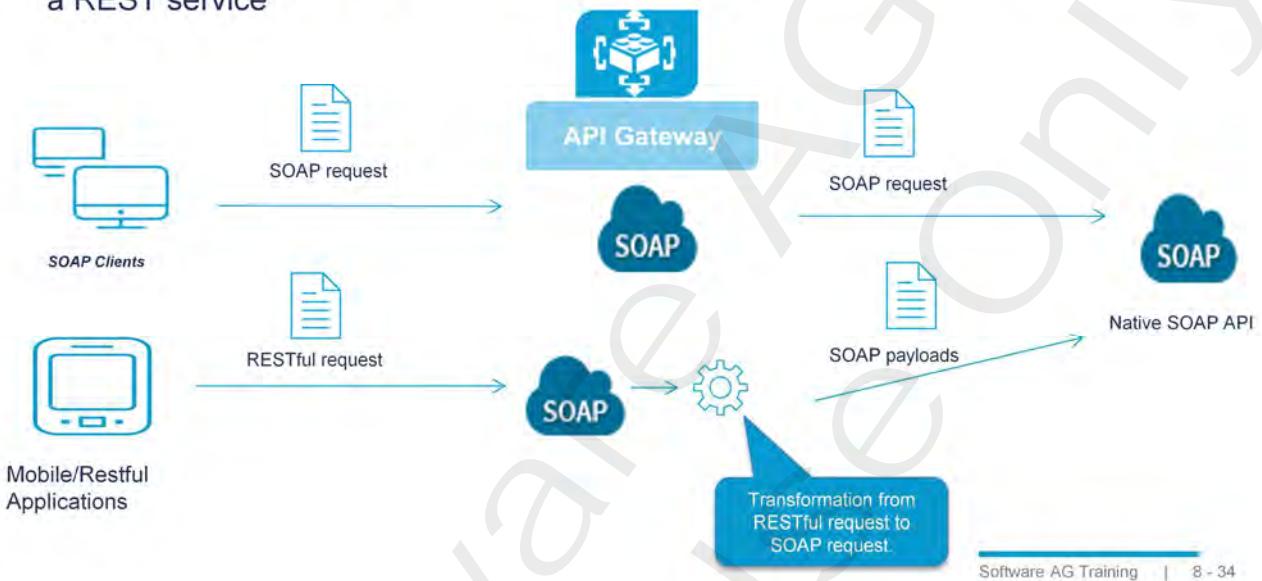


SOAP to REST Transformation

Notes:

Use Case for SOAP to REST Transformation

- SOAP APIs are available to expose the application inside the enterprise and to external partners / consumers
- Going towards REST APIs (mobile apps, device integration,)
 - Consumers expect a REST interface instead of sending SOAP messages
 - Transformation from SOAP to REST provides this without recreating the service as a REST service



Software AG Training | 8 - 34

Notes:

SOAP To REST Transformation Capabilities in API Gateway

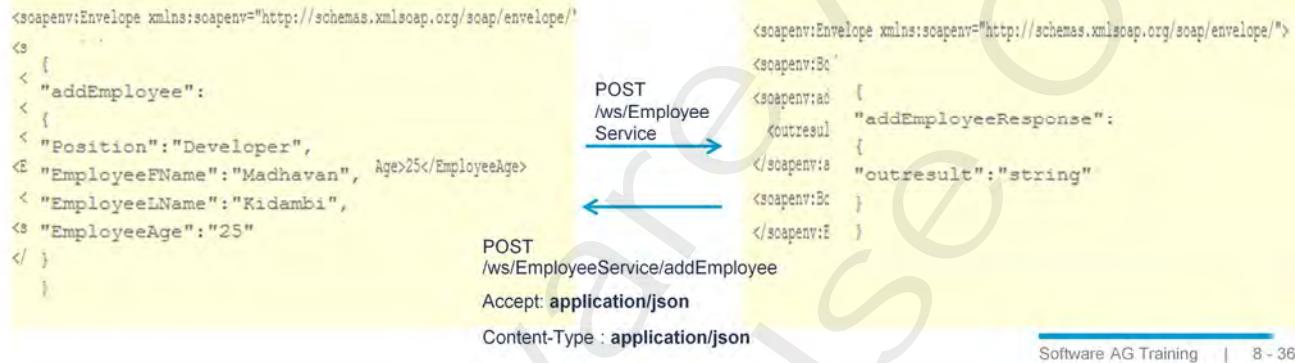
- REST Transformation option is supported for all SOAP APIs in API Gateway
 - Disabled by default
 - Provider can enable all/some operations of an SOAP API to be exposed as REST to the users.
- After enabling the Transformation
 - SOAP Operation is available a REST Resource / Method combination set up with default values.

The screenshot shows two panels side-by-side. The left panel is titled 'Operations' and lists an operation named 'addEmployee'. Below it, a green button labeled 'SOAP11' is followed by the text 'SOAPnativeService_employeeService_addEmployeeES_WS...'. The right panel is titled 'REST transformation' and lists the same operation 'addEmployee - /addEmployee'. A large blue curved arrow points from the 'Operations' panel to the 'REST transformation' panel, indicating the transformation process. At the bottom right of the interface, there is a footer with the text 'Software AG Training | 8 - 35'.

Notes:

SOAP To REST Transformation Capabilities in API Gateway

- When the transformation for a SOAP operation is enabled , the operation can be invoked RESTfully with default values.
- The Default resource path is the name of the SOAP Operation and default HTTP Method is POST.
- The API Provider can configure custom values for the resource paths and the methods.
- The Provider can also modify the default request/response schemas and examples that are generated.



Notes:

Processing Path and Query parameters

- Path parameters in a resource path and query parameters in the request need to be mapped to the SOAP element in request to be sent to the Native service using Xpaths.

```

    GET /ws/EmployeeService/employee/{employeeID}
    /employee/ID1
    
```

```

    <Envelope>
      <Body>
        <getEmployee>
          <InputID>ID1</InputID>
        </getEmployee>
      </Body>
    </Envelope>
    
```

- Mapping needs to be defined for the path parameter

employeeID						
Name *	Description	Type	Data type	Required	Repeat	Value
employeeID	The Employee ID to be used	Path	String	Required	Repeat	Value
XPath		Namespace prefix *		Namespace URI *		
/InputID						
<input type="button" value="update"/>						

Software AG Training | 8 - 37

Notes:

Software AG Training | 8 - 38

Advanced capabilities

- API Gateway allows the REST definition of the SOAP API to be exported as a Swagger/RAML files for the external clients.

 Specifications

Artifacts	EmployeeService.zip
Swagger data	EmployeeService.json
RAML data	EmployeeService.raml

Swagger and
RAML data
available in the
Specifications

- SOAP attachments handling
 - SOAP APIs handling attachments can be designed as RESTful interfaces accepting multipart/form-data or multipart/mixed data.
 - API Gateway takes care of converting the different MIME parts into SOAP attachments.

Notes:



9

Analytics in API Gateway

Notes:

Objectives

At the end of this chapter you ...

- Know about the different dashboard capabilities in API Gateway

Notes:

Analytics in API Gateway

- Analytics gives Information about
 - API Gateway events
 - API-specific events
 - API trends (which APIs are more popular)
- Widgets are grouped contextually together and placed as dashboards.
- Pictorial and Textual widgets are available to
 - Easily Understand
 - Analyze and compare
 - Act based on the generated data
- API Gateway support two different types of analytics
 - API Analytics -> API Gateway Provider
 - Gateway Analytics -> API Gateway Administrator



Software AG Training | 9 - 3

Notes:

API Gateway Dashboards - Data

- Events
 - generated based on the policies/transactions in real time
 - Persisted in internal data store (Elasticsearch)
 - 
- API Gateway Analytics
 - renders the data present in the local storage as a variety of charts => different dashboards (Kibana)
 - Filtering allows aggregation of data over a custom period of time
 - Discover patterns
 - Drill down into invocation details

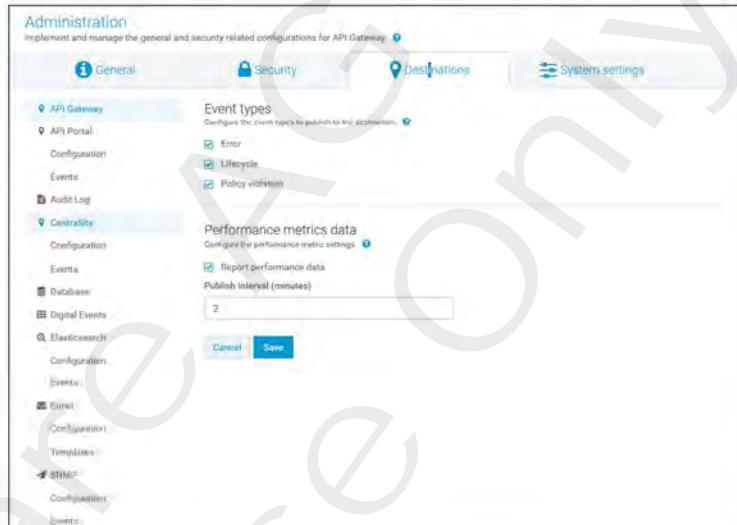


Software AG Training | 9 - 4

Notes:

Event Configuration

- Most events are policy driven
 - Destination can be configured within the policy
- Others can be controlled by API Gateway Administration > Destinations
 - Event Types can be disabled
 - Performance data can be enabled
 - Specify publish interval



Software AG Training | 9 - 5

Notes:

Event Types in API Gateway

- Runtime Events
 - For most a corresponding Policy action needs to be configured

Event Type	Description
Transaction	Is generated each time a virtual service is invoked. Associated with Log Invocation Policy
Error	Is generated each time an error occurs during a runtime service invocation.
Monitoring	Is generated when a threshold is exceeded in any of the configured parameters in the Monitoring Policy. Monitoring can be done on application level so that each consumer can be tracked individually.
Policy Violation	Is generated to alert the provider when a policy associated with an API is violated.
Lifecycle	Is generated whenever the instance is started or stopped.
Threat Protection	Is generated whenever a threat protection policy is violated.
Performance Metrics	Is generated for every API for a specific interval. Provider or Administrator can configure this interval in the destination configuration page. The event provides information about total invocations for the API in the specified interval along with the response time calculations ...

Notes:

Dashboards – API Gateway Analytics

- Provides information about the overall system behaviour, trends and activities via visualizations
- Available dashboards
 - Summary
 - Trends
 - Application
 - Packages
 - Threat Protection
 - Archive and purge
 - Restore

Notes:

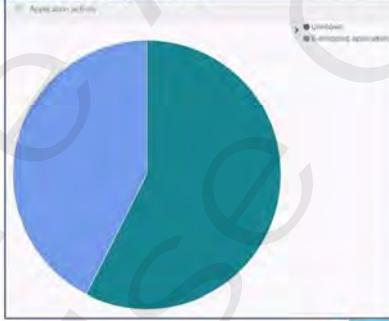
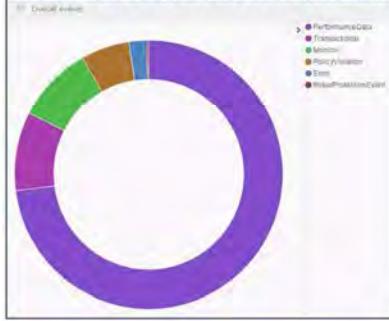
Dashboards – API Analytics

- Provides information about individual API invocations
- Available dashboard
 - Events over time
 - API invocations
 - API invocation pattern
 - Native service performance
 - Response code trend
 - API trend by response
 - Success vs Failure
 - Runtime events
 - Service result cache
 - Method level invocations

Notes:

Global Dashboard - Summary

- Overall Events
 - Different category of events
- Application Activity
 - Activities of different applications calculated by the number of invocation to their corresponding APIs
- Runtime Events
 - Events in text form => better understanding and analysis
- Payload Size
 - Top APIs transferring higher volume in request/response

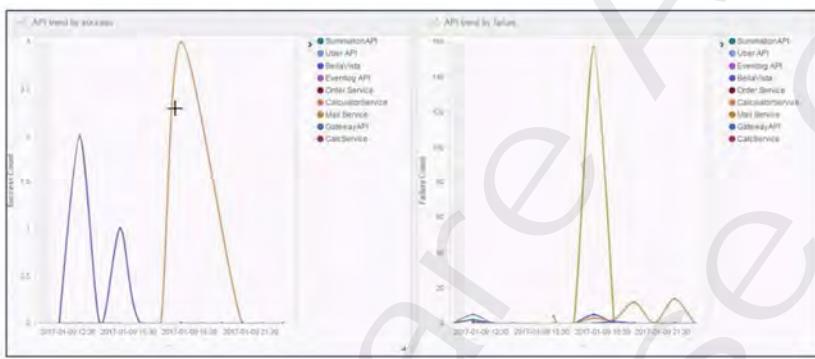


Software AG Training | 9 - 9

Notes:

Global Dashboard - Trends

- Events Over Time
 - Different types of events over time
- API Trends by Success
 - Trending APIs based on success
- API Trends by Success
 - Trending APIs based on failure rate
- Overall Error Trend
 - Overall error trend of the system



Software AG Training | 9 - 10

Notes:

Global Dashboard - Applications

- Events per Application
 - Events related to application-specific invocations of the API. Application could be a filter.
- Violation per Application
 - Events where the application is involved
- Activity Rate for Consumed APIs
 - Number of invocations where the application is used to invoke the API
- Payload Size
 - Top APIs transferring higher volume in request/response

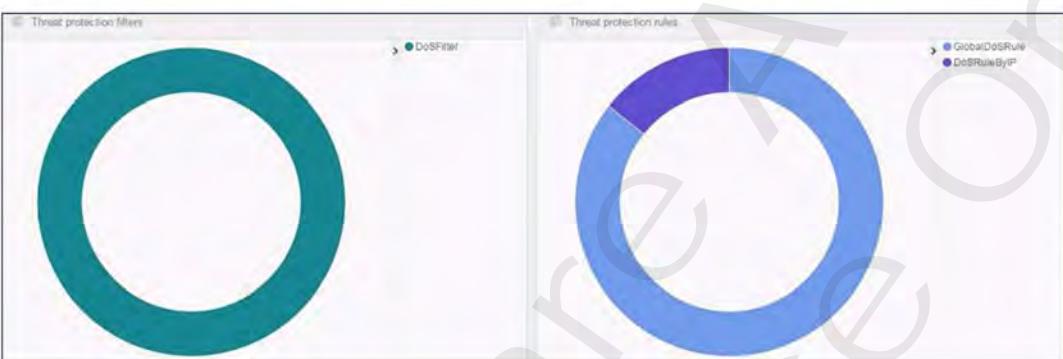
The dashboard displays two main sections: 'Events per application' and 'Violations per application'. The 'Events per application' section shows a stacked bar chart for the 'E-shopping application' across four time intervals. The 'Violations per application' section shows a stacked bar chart for the same application across three time intervals. Both charts use a color-coded legend: blue for 'Normal', red for 'Error', and purple for 'Panic/Violation'.

Software AG Training | 9 - 11

Notes:

Global Dashboard – Threat Protection

- Threat Protection Rules
 - Threat protection rules which are violated
- Threat Protection Filters
 - Threat protection filters which are violated
- Threat Protection Events
 - Events in text form for better understanding and analysis



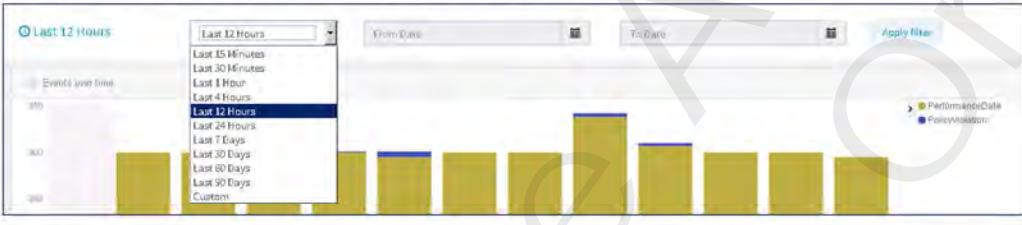
The dashboard displays two donut charts. The left chart, titled 'Threat protection filters', shows a single large teal segment representing 100% of the filters. The right chart, titled 'Threat protection rules', shows two segments: a blue segment labeled 'GlobalDoSRule' and a purple segment labeled 'DoSRuleByIP', with the total rule count being 100%.

Software AG Training | 9 - 12

Notes:

Custom Filters

- Custom filters are available to filter and visualize the data efficiently
 - Once the filter is selected/applied all widgets will be adapted to current filter settings
- 2 ways to do filtering
 - Predefined time intervals
 - Predefined frequently used time intervals
 - Ease use of access
 - Custom time selection
 - User defined From and To dates
 - Analyze data for some specified interval of time



The screenshot shows a dashboard interface with a bar chart and a filter panel. The filter panel on the left has a dropdown menu for selecting a time interval. The menu is open, showing options like 'Last 12 Hours', 'Last 24 Hours', 'Last 7 Days', etc., with 'Last 12 Hours' highlighted. There are also buttons for 'From Date' and 'To Date' and an 'Apply filter' button. The main area shows a bar chart with several bars, some colored yellow and one blue, representing different data series over time.

- Interactive filtering
 - Within the dashboards select a specific property to drill down

Software AG Training | 9 - 13

Notes:

Purging of Data

- Data Management
 - Purging of data is supported for data stored in the internal store (Elasticsearch)
- 2 kinds of purging, define by the API Gateway Administrator:
 - Date
 - Purge data till a specified date
 - Example: 2017-01-01
 - System removes all analytical data available till this date
 - Duration
 - Purging data for a specified time interval
 - Example: 6M

Software AG Training | 9 - 14

Notes:



10

API Portal Overview

Notes:

Objectives

At the end of this chapter you ...

- Get an overview of API Portal capabilities
- Know how to integrate API Portal and API Gateway
- Understand user management in API Portal

Notes:

Chapter Contents

- API Portal Administration
- User Management

Notes:

Business Objectives for API Programs

- New revenue streams
 - Introduce new Sales channels
 - Selling products through partner eCommerce sites and apps
- Better visibility
 - Gain broader visibility on the web, allow your customers to find information they are looking for
- Stronger community
 - Increased community adoption of your services
- Improved customer experience
 - Enhanced user experience with accurate and relevant data for better loyalty
- Partners
 - Automated B2B partner transactions



Software AG Training | 10 - 4

Notes on Customer Stories for Each of these Objectives

New Revenue Streams = Dick's Sporting Goods

New Products = Outerwall/Redbox

Better Visibility = Outerwall/Redbox (appear on Walgreen's website), EDF

Stronger Community = AXA

Improved Customer Experience = Comcast, LEGO, Echo Entertainment

Partners = Yale

Benefits of an API Strategy

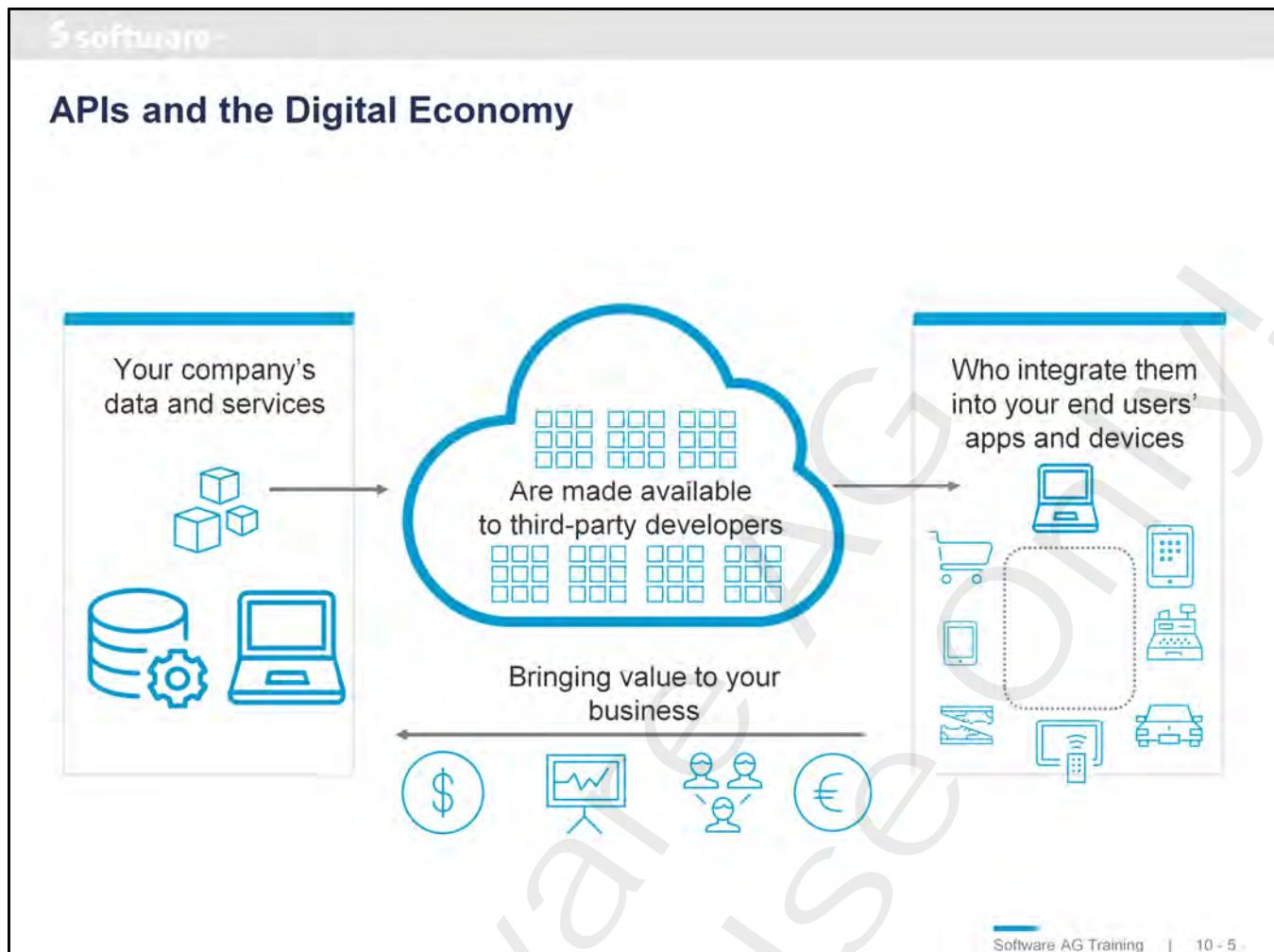
APIs are key to digital transformation, and that is the goal. APIs aren't the goal in and of themselves

Agility - Build new services more quickly

Improved real time data for improved customer experience

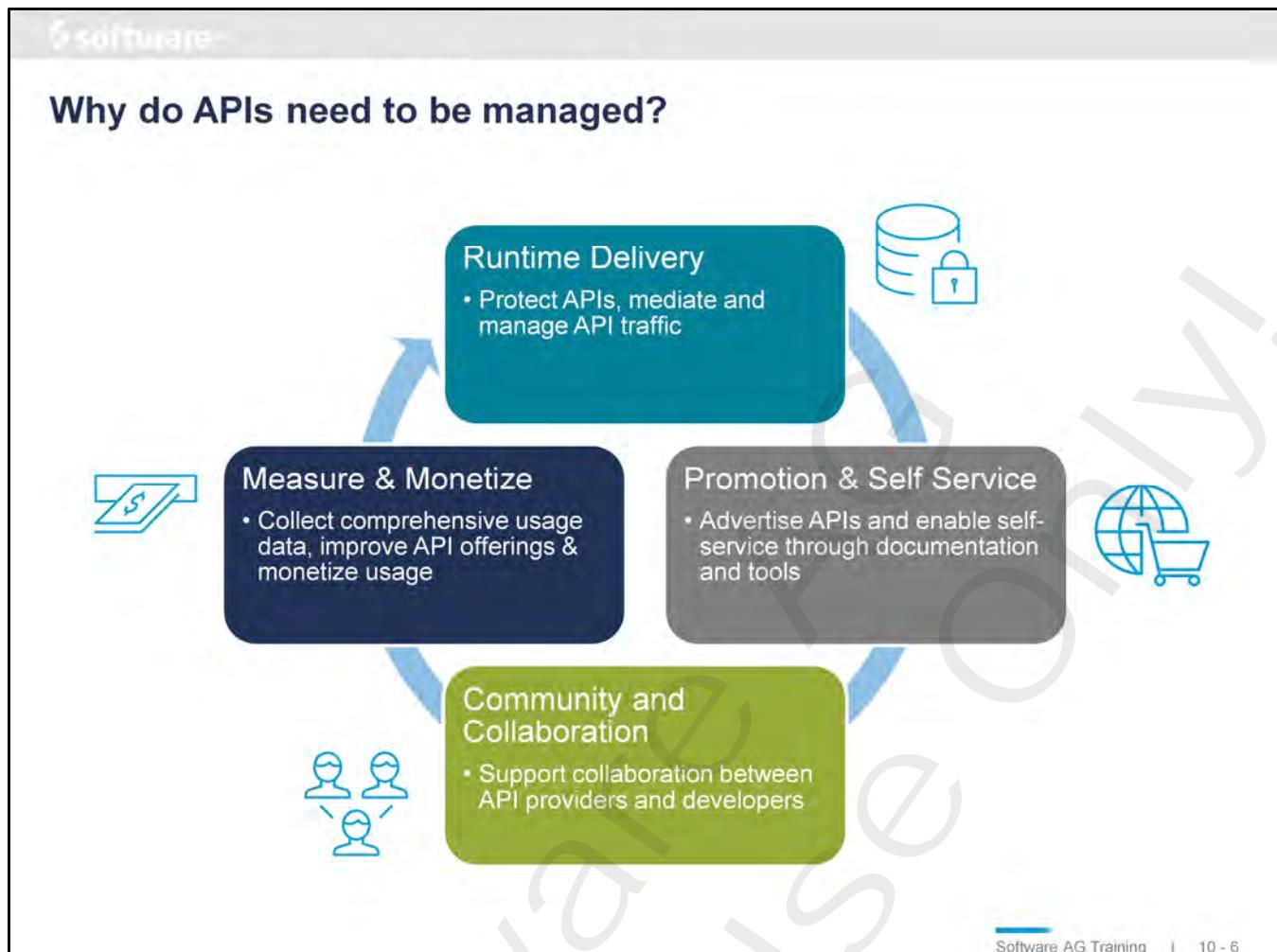
Industry standard access to data/services

Abstract underlying legacy apps to make standard APIs - for better manageability



Software AG Training | 10 - 5 -

Notes:

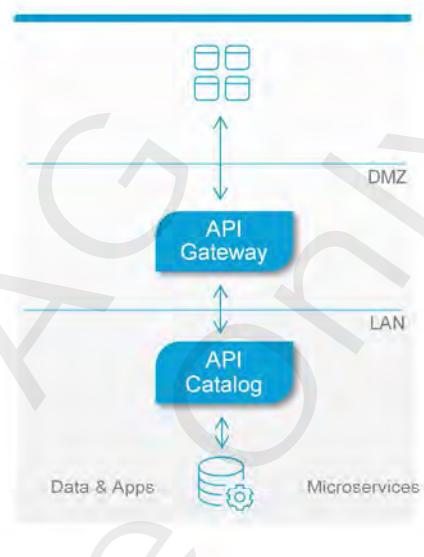


Software AG Training | 10 - 6

Notes:

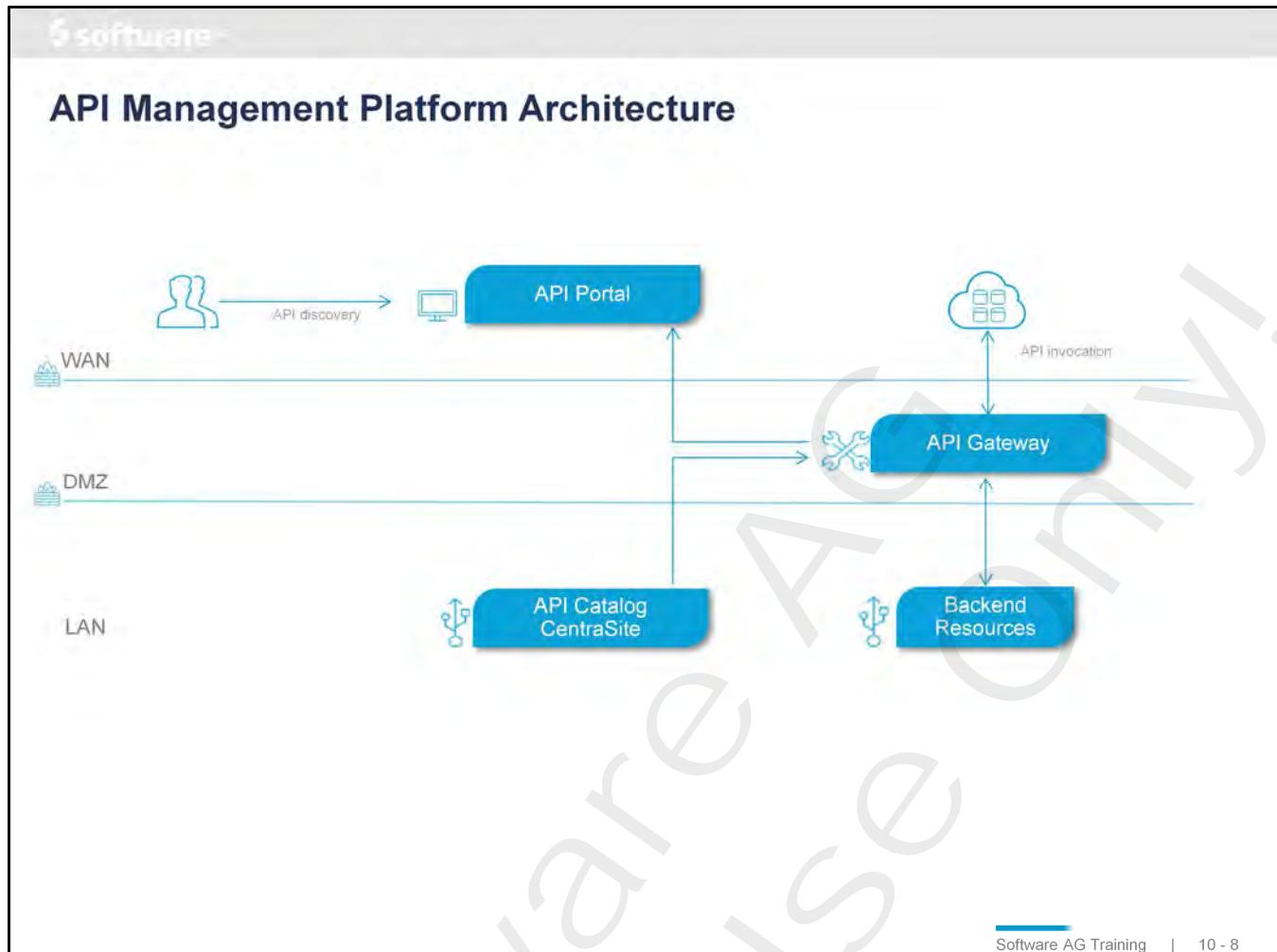
API Gateway: secure and protect APIs

- API Gateway
 - API Documentation
 - DMZ level protection
 - API Mediation
 - Analyze API runtime usage
- API Catalog
 - Define your APIs
 - Manage Lifecycles
 - Track dependencies



Software AG Training | 10 - 7

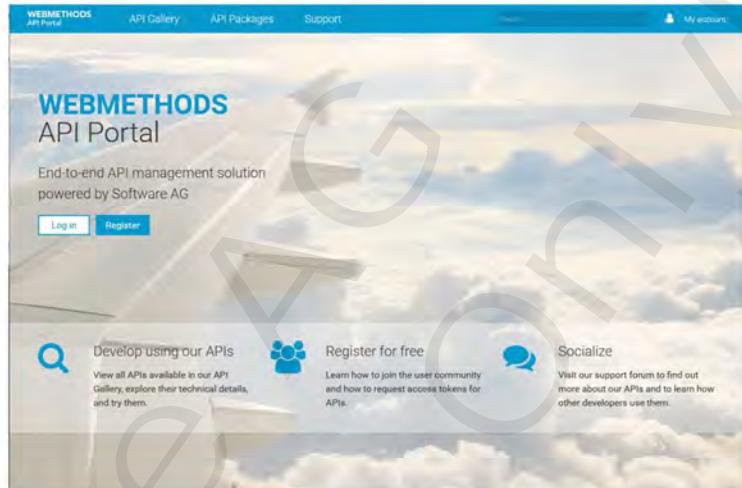
Notes:



Notes:

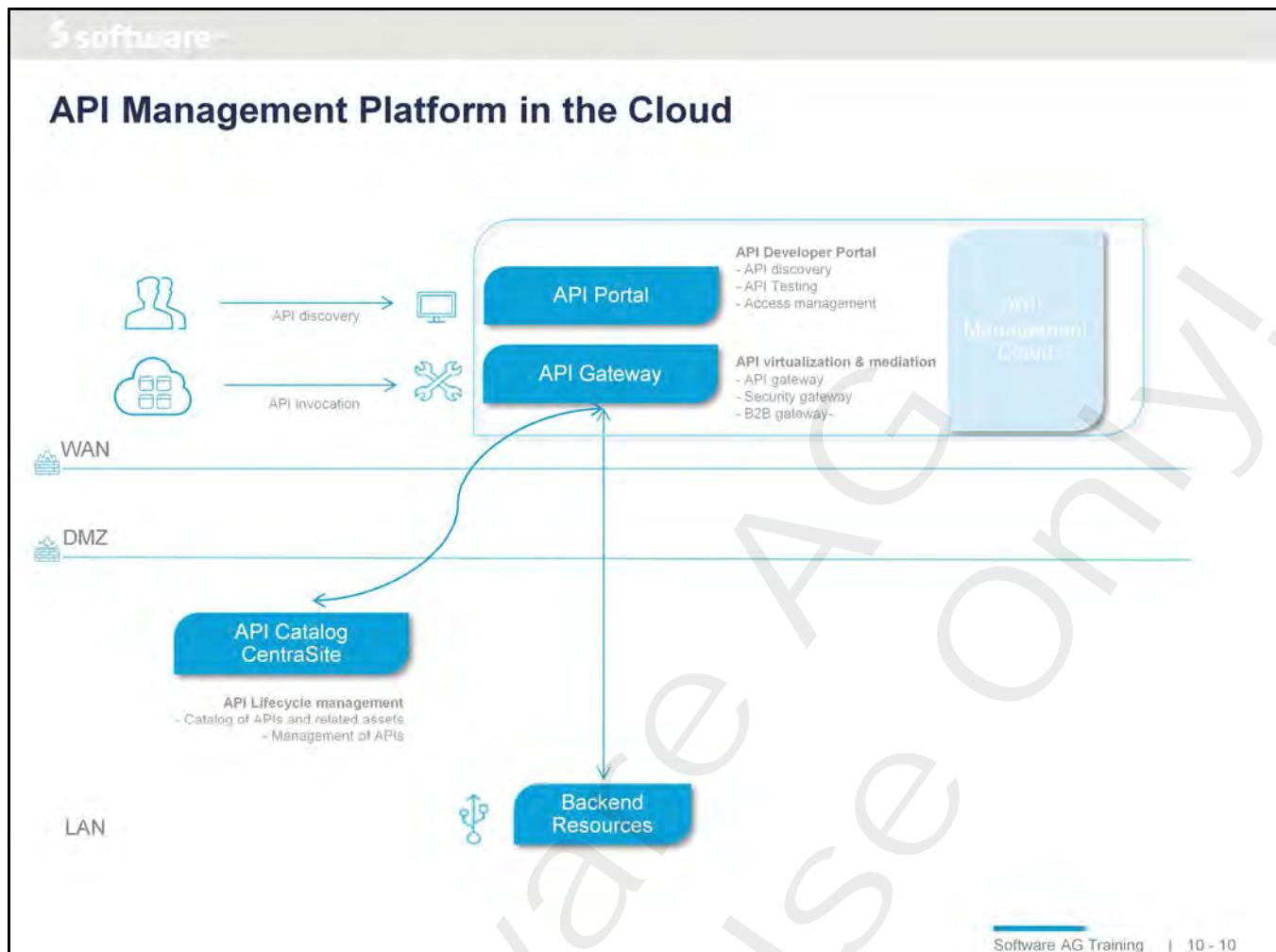
API Portal

- Self-service portal to expose APIs for developers and B2B partners
- Build-in Usage Analytics
- REST and SOAP API Testing
- Metadata driven API Documentation
- Integrated Collaboration
- Private API Communities
- Multi Tenant Architecture
- Tenant backup & restore



Software AG Training | 10 - 9

Notes:



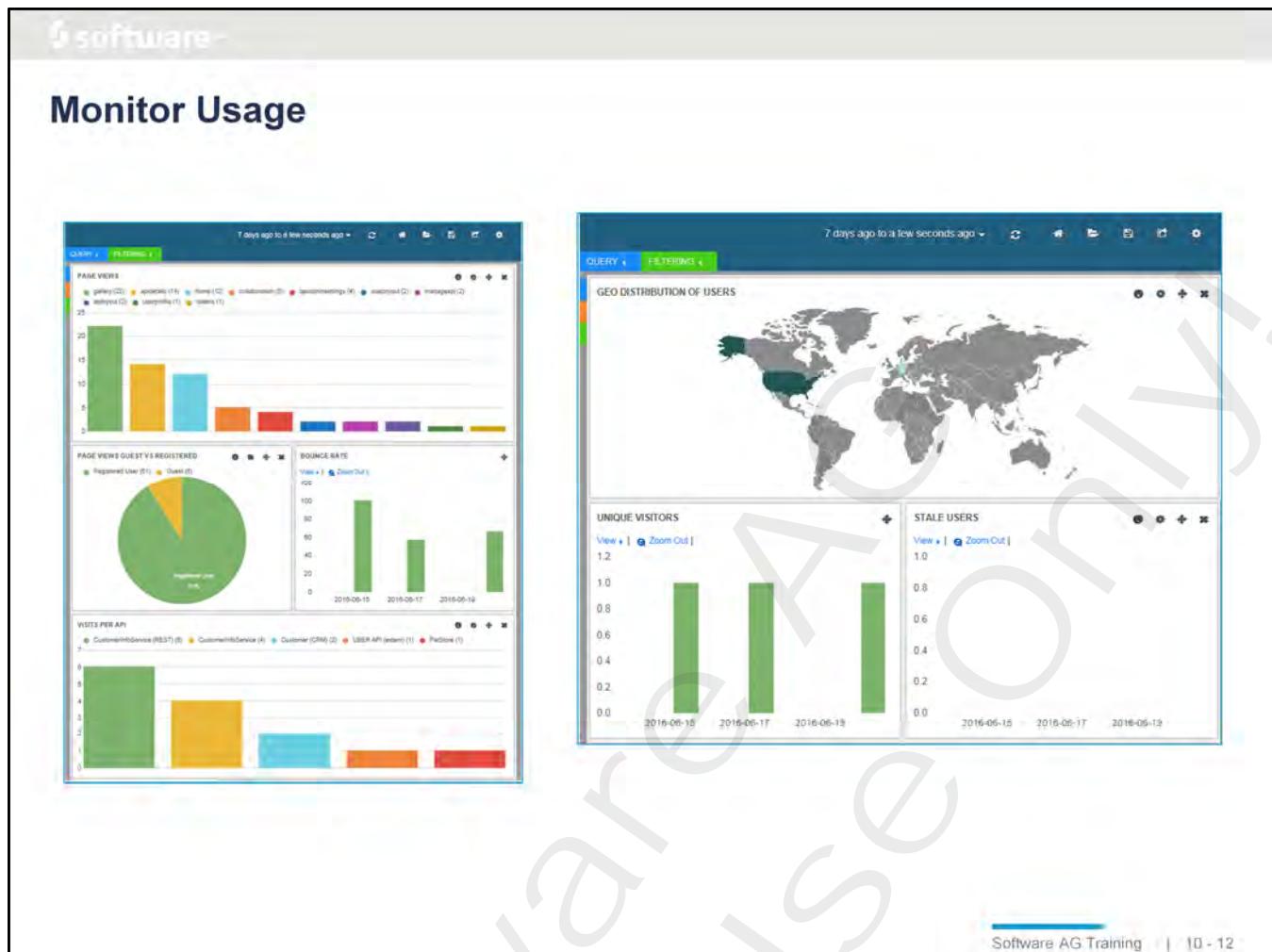
Notes:

Overview

- API Provider exposes APIs to external Developers
 - The developers will build their own applications using these APIs
 - These applications will be consumed by end users
- Providing APIs to external Usage
 - For free
 - generate revenue through various business models chosen by the API Provider
 - Quality of Service must somehow be guaranteed / monitored
- Monitoring & measuring the performance of the API
 - To know the success of an API or to claim that the API is profitable
 - Identify better avenues for revenue generation using existing data
 - Understand the consumers' interests and introducing attractive plans and services

Software AG Training | 10 - 11

Notes:



Notes:



API Portal Administration

Notes:

Configuring API Gateway and API Portal

- Create API Provider users (developers) in API Gateway who can define APIs
- Create a user in API Gateway who has assigned the publish to API Portal functional privilege
- Create API Gateway Administrator users in API Gateway who can on behalf of API Portal create applications, request or invoke access tokens, subscriptions, ...publish the APIs to API Portal
- Create API Provider user in API Portal who can on behalf of API Gateway create APIs in API Portal
- Configure API Portal destination in API Gateway
- Set up and customize email templates to be used when a consumer in API Portal requests an application

Software AG Training | 10 - 14

Notes:

How to Integrate the API Gateway and API-Portal?

- API documents (descriptions) are automatically generated from **API Gateway** metadata
 - No duplicate maintenance of API descriptions
 - APIs are published directly from API Gateway
- **API Gateway and API Portal Integration**
 - Wiring will be done ONLY in API Gateway
 - Available in the Destination within API Gateway Administration

The screenshot shows the 'Administration' section of the webMethods API Gateway. The top navigation bar includes links for 'APIs', 'Policies', 'Applications', 'Packages', and 'Destinations'. A blue arrow points to the 'Destinations' link, which is highlighted with a blue border. Below the navigation, there's a sub-navigation with tabs: 'General', 'Security', 'Destinations' (which is active), and 'System settings'. The main content area is titled 'Administration' with the subtitle 'Implement and manage the general and security related configurations for API Gateway.' There are also 'Home' and 'Administrator' links at the top right.

Software AG Training | 10 - 15

Notes:

Creating API-Portal as API Gateway Instance

- In Destination select **API Portal > Configuration**
 - Provide basic information for this API Portal instance
- For both directions of communication we need a Technical User
 - API Gateway: to be used for API Portal -> API Gateway communication
 - API Portal: to be used for API Gateway -> API Portal communication

The screenshot shows the 'Administration' interface for managing configurations for API Gateway. The top navigation bar includes tabs for General, Security, Destinations, and System settings. Below the tabs, there are two main sections: 'API Gateway' and 'API Portal'. The 'API Portal' section is expanded, revealing sub-options like 'Communication' (which is selected and highlighted with a blue border) and 'Configuration'. A sub-section titled 'API Portal communication' is visible, with the sub-option 'Basic information' selected. On the right side of the interface, there is a 'Name' input field. The overall layout is clean and organized, typical of a web-based administration tool.

Software AG Training | 10 - 16

Notes:

The screenshot shows the Software AG API-Portal configuration interface. The left sidebar lists 'Administration' (General, Gateway, Database, API Portal), 'Configuration' (Events), and a 'Portal configuration' section. The main area has two tabs: 'Portal configuration' (selected) and 'Gateway configuration'. The 'Portal configuration' tab contains fields for Base URL (http://localhost:18101), Tenant (default), Username (Andy), and Password (*****). The 'Gateway configuration' tab contains fields for Base URL (https://daetrain00749.eu.ad.sag:5555), Username (Andy), and Password (*****). A blue callout box labeled 'Publish API-Portal instance!' points to the 'Publish' button in the bottom-left corner of the configuration tabs. Two blue speech bubbles provide communication setup details:

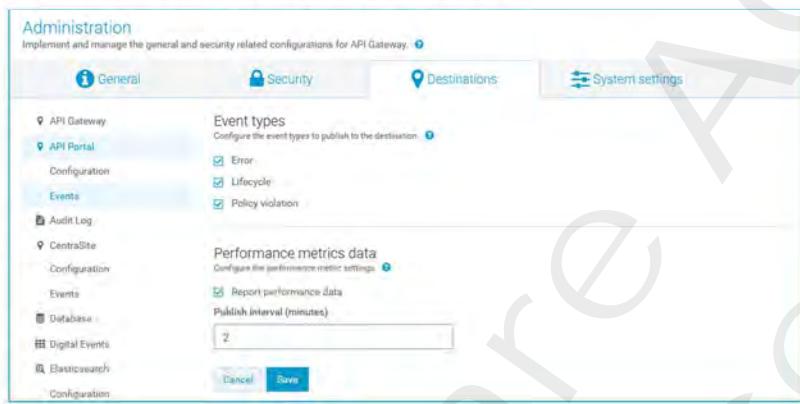
- Communication Setup for API Gateway talking to API-Portal instance**
 - API Portal Endpoint
 - Tenant
 - User in API Portal

User role required in API-Portal:
API-Provider
- Communication Setup for API-Portal talking to API Gateway**
 - API Gateway Endpoint
 - User does NOT need to have API Administrator role

Notes:

API Portal Configuration for Events

- API Portal configuration for Events/Metrics in Administration
 - The Event types & Performance Metrics configured on this page are common for the API Portal destinations listed below.
- API Portal instances are populated when an API with API Portal destination is published from API Gateway.
 - This will be covered later in the course



The screenshot shows the 'Administration' interface with the 'Events' tab selected. Under 'Event types', there are three checked checkboxes: 'Error', 'Lifecycle', and 'Policy violation'. Under 'Performance metrics data', there is a checked checkbox for 'Report performance data' and a dropdown menu set to '2'. At the bottom are 'Cancel' and 'Save' buttons.

Software AG Training | 10 - 18

Notes:



User Management

Notes:

User roles for API-Portal and their primary tasks



- API-Provider

- Publishing of API-Documentations from CentraSite
- Review of published API-Documentations
- Support of their APIs in support forums



- API-Portal Administrator

- Setup and Configuration of Infrastructure (API-Portal, CentraSite, Mediator)
- Housekeeping & daily operations
- Portal Customization
- Monitoring usage



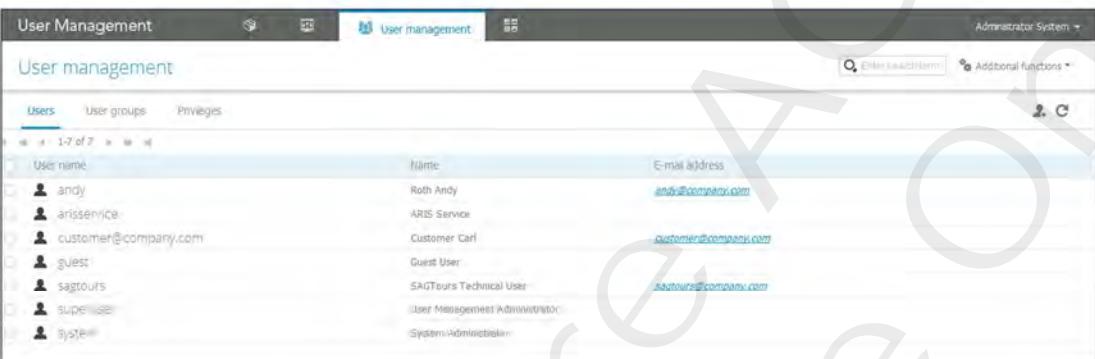
- API-Consumer

- Discovery of APIs
- Review of API details
- Testing of APIs
- Self on boarding
- Requesting access tokens

Notes:

Manage Users in the API-Portal

- User management is provided by the UMC component
 - Tenant aware
 - Reachable as `http://<host>:18101/umc`
 - Login as user: `system / manager` (**NOT** Administrator / manage)



The screenshot shows the 'User Management' interface with the 'User management' tab selected. The left sidebar has 'Users' selected. The main area displays a list of users with their names and email addresses:

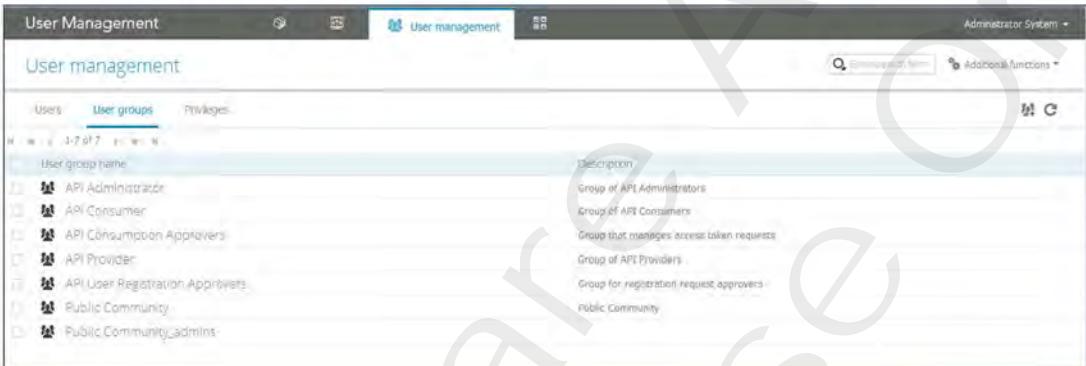
User name	Name	E-mail address
andy	Roth Andy	andy@company.com
arisservice	ARIS Service	customer@company.com
customer@company.com	Customer Carl	customer@company.com
guest	Guest User	customer@company.com
sagtours	SAGTours Technical User	customer@company.com
superuser	User Management Administrator	customer@company.com
system	System Administrator	customer@company.com

Software AG Training | 10 - 21

Notes:

Managing Groups in API-Portal

- Default groups created during installation within UMC
 - API Administrator – for purely Administration
 - API Consumer – for consumer self onboarding, they will be added automatically here
 - API Consumption Approvers – managing access tokens requests
 - API Provider – Administrator needs to add all users that will publish APIs
 - API User Registration Approvers – responsible for approve registration requests



The screenshot shows the 'User Management' interface with the 'User groups' tab selected. On the left, there's a tree view of user groups. On the right, there's a detailed view of each group with its description. The groups listed are:

User group name	Description
API Administrator	Group of API Administrators
API Consumer	Group of API Consumers
API Consumption Approvers	Group that manages access token requests
API Provider	Group of API Providers
API User Registration Approvers	Group for registration request approvers
Public Community	Public Community
Public Community Admins	

Software AG Training | 10 - 22

Notes:

API Portal User Management

- Web based Administration
- Interface to manage users, groups and permissions
- Internal user store as default
- Integration with LDAP optional
- Integrated usage dashboard

The screenshot shows the 'User Management' dashboard with the following details:

- License usage:** A bar chart titled "weMethods API Portal license" showing usage over time. The Y-axis ranges from 0 to 1000, and the X-axis shows dates from 01.01.2016 to 28.02.2016. The chart shows a single bar reaching approximately 1000 units.
- Users:** Shows 3 active users and 7 user groups.
- Last LDAP synchronization:** Shows the last sync was at 10:00 on 2016-02-28.
- Users online:** Shows 0 users currently online.
- Logins within the last 24 hours:** A bar chart showing the number of logins per hour. The Y-axis is "Number of logins" (0-2) and the X-axis is time (17:00, 18:00, 19:00, 20:00, 21:00, 22:00, 23:00). One bar is visible at 18:00.

Software AG Training | 10 - 23

Notes:



Exercise 15

- Setting up Integration with API Portal Objectives

Notes:



11

API Portal as Provider

Notes:

Objectives

At the end of this chapter you ...

- How to configure your API for API Portal
- How to publish the APIs to API Portal
- How to use API Portal as API Provider

Notes:

Chapter Contents

- Publishing APIs to API Portal
- Lifecycle Control
- API Import in API Portal

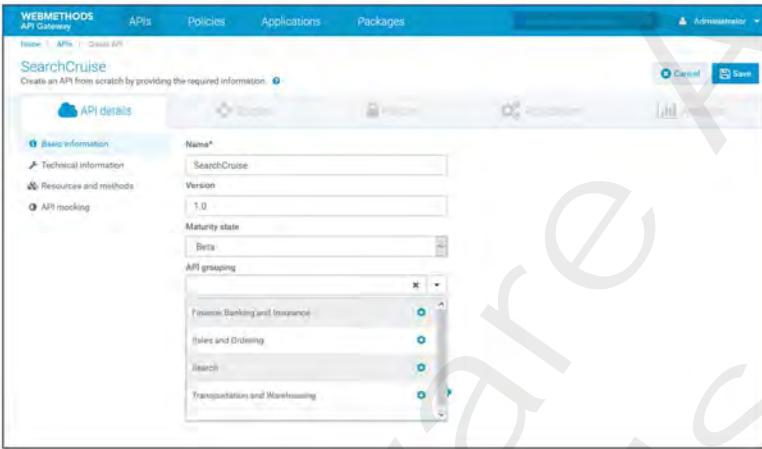


Publishing APIs to API Portal

Notes:

Categorizing APIs: API Grouping

- Classify APIs using a definable criteria
 - Help developers to discover APIs in larger catalogs
- Taxonomies can be defined in API Gateway
 - API Grouping (CRM, Finance, Sales and Ordering, Search, Transportation, ...)
 - API Maturity Information (Beta, Experimental, Production, Test)
 - These are predefined and are customizable by Administrator in API Gateway

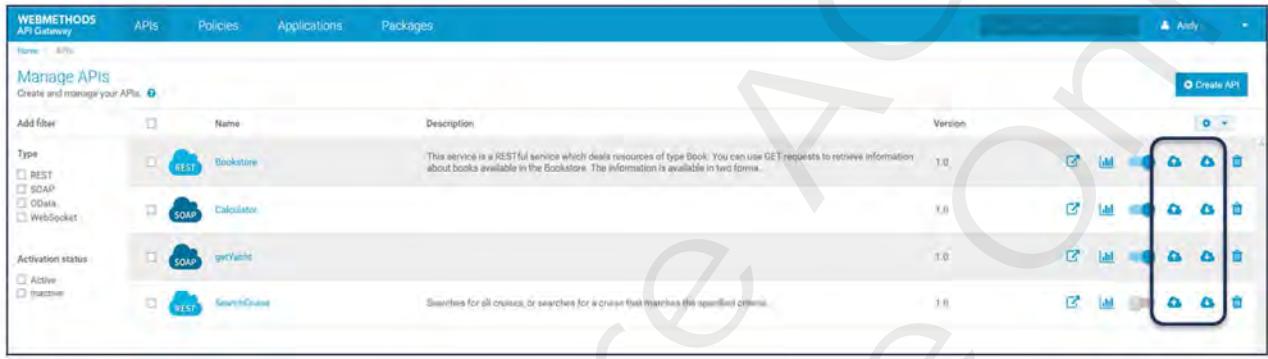


Software AG Training | 11 - 5

Notes:

Publishing API to API-Portal

- **Publish/Unpublish** action to API-Portal is available on the API details page
- Pre-Requisites
 - API Gateway role **API Gateway Administrator** to publish to API-Portal instances
 - Registered **API-Portal** instance



Software AG Training | 11 - 6

Notes:

Software AG

Selecting Properties for Publishing to API Portal

- Select for target based on
 - Active API Endpoint available on API Gateway
 - Communities defined in API Portal

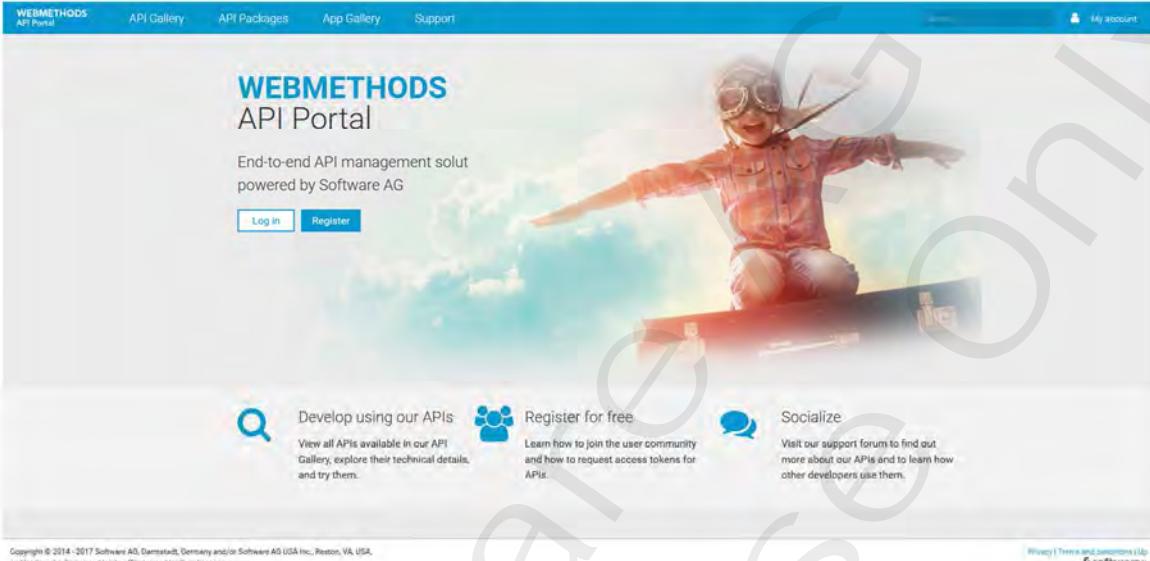
Name	Description
Public Community	Public Community

Software AG Training | 11 / 7

Notes:

APIs in API-Portal

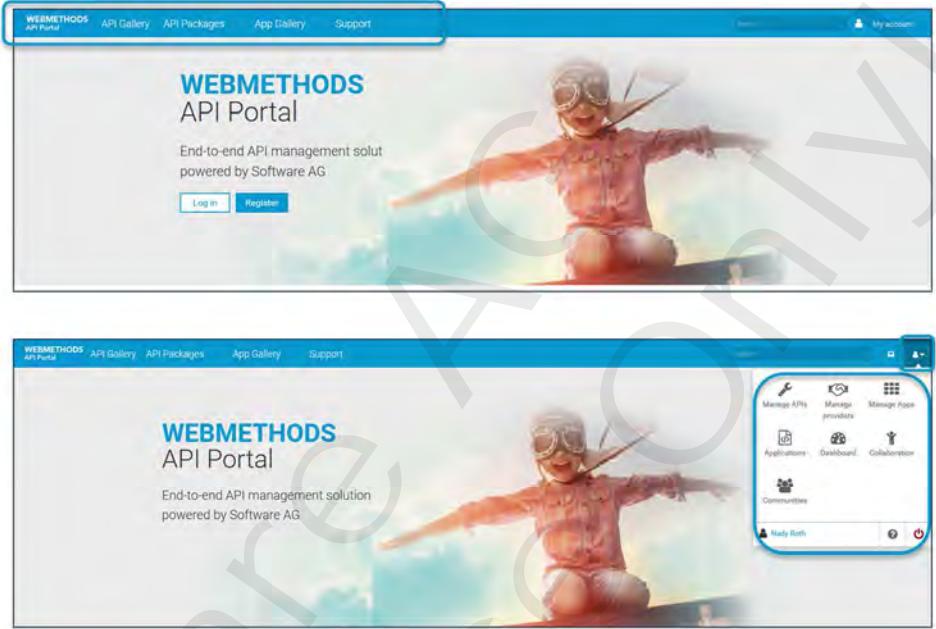
- Logon to API-Portal using URL
 - <http://<hostname>:18101/#default/home>
 - Defaulted to API Portal home page, API Landing Page



Notes:

API-Portal Pages and Navigation

- Header Navigation – Role based
 - Home (Default)
 - API Gallery
 - API Packages
 - App Gallery
 - Support
- Additional Options after login
 - Communities
 - Dashboard
 - Administration
 - ...



Software AG Training | 11 - 9

Notes:

Searching API using Find

- Searching for APIs
 - simple keyword search
 - to search directly for a particular API
 - considers API-Name AND all details of the API
 - search matches where API title doesn't match, but description does

The screenshot displays the 'API Gallery' section of the webMethods API Portal. At the top, there's a navigation bar with links for 'WEBMETHODS API Portal', 'API Gallery', 'API Packages', 'App Gallery', 'Support', and 'My account'. Below the navigation bar, the URL 'Home > API Gallery' is shown. A search bar with the placeholder 'Search' is positioned above the main content area. To the right of the search bar is a dropdown menu labeled 'Grouped by'. The main content area shows two search results pages. The first page has a search term 'Get' and displays 11 matches in categories such as 'Bookstore', 'REST API', and 'GET'. The second page has a search term 'GetYacht' and displays 6 matches in categories such as 'SOAP API', 'SOAPAction', and 'API parameter'. Both pages include a 'Show all' link at the bottom.

Notes:



Advanced Searching

- Search by keyword
- Type-ahead hints
- Comprehensive search results screen
- Add filters to refine search
- Save search as favorite

The screenshot shows the 'API Gallery' section of the webMethods API Portal. A search bar at the top contains the text 'GetYacht'. Below it, a message says '6 matches'. The results list includes:

- GetYacht - SOAP API (Path /GetYacht)
- SOAPAction - API parameter
- getYachtFromDB - SOAP operation
- (Untitled) - SOAP messages
- http://daelittrain26233:5555/ws/getYacht/1.0 - End points
- http://daelittrain26233:5555/ws/getYacht/1.0 - End points

Software AG Training | 11 - 11

Notes:

WEBMETHODS API Portal

API Gallery API Packages App Gallery Support My account

Home | API Gallery

API Gallery

Customer Management

SignupAPI

REST

API for signing-up to SAGTours. The signing-up resources creates/updates/deletes a customer from SAGTours. Sign-up and Booking are protected via Basic Authentication. The customerID is the email address. If an error occurs, the Sign-up API may give a text message as well as the HTTP response code.

PRODUCTION

[View details >>](#)

Sales and Ordering

BookingsAPI

REST

You found the perfect cruise, now lets get it booked. Before a cruiser can be booked, the customer must sign up to the SAGTours site if they have not already done so. Once signed-up, use the Bookings API to book a cruise. You need the credentials you gave during sign up to book the cruise. The credentials should be sent to the API using HTTP Basic.

PRODUCTION

Grouped by

- API group
- Business term
- Maturity status

- **API Gallery** allows simple browsing for available APIs
- APIs can be **ordered by**
 - Maturity level
 - Business Terms
 - API Groups
- Ordering criteria is based on published metadata (**API-Portal Information profile**)

Software AG Training | 11 - 12

Notes:

Metadata driven API Docs

- APIs are published directly from CentraSite
- API Gallery is dynamically rendered based on metadata definitions
- API-Portal documentation is automatically generated from rich CentraSite API metadata
 - No manual re-documentation of the API for consumption in the API Portal
- Provider and Consumer view on API stay in sync
- Internal and external consumers share the same API documentation



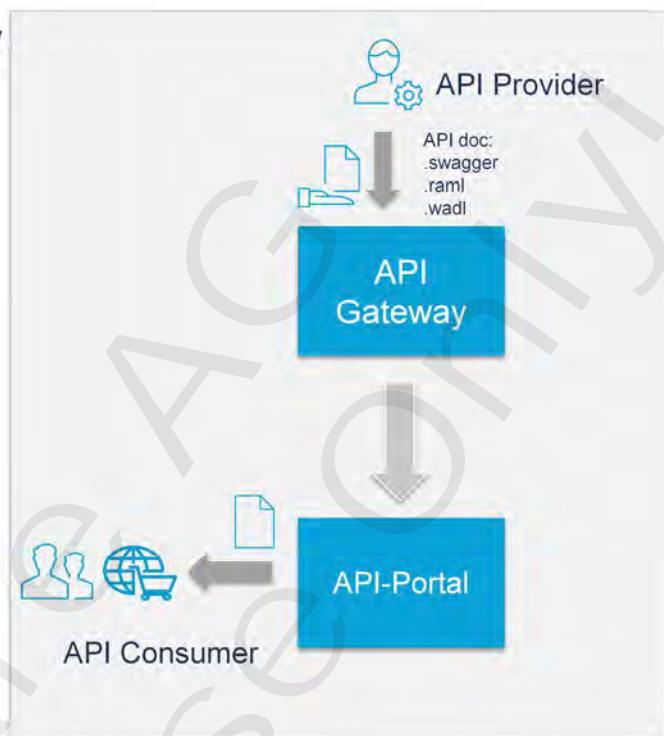
Software AG Training | 11 - 13

Notes:

Documenting REST APIs

- Creation of REST APIs in API Gateway
 - Import based on
 - Swagger, RAML,
 - Manual API documentation
 - Creation of SOAP based APIs
 - Import based on WSDL

```
  "swagger": "2.0",
  "info": {
    "description": "This is a sample server Petstore server. You can find out more about Swagger at <a href=\"http://swagger.wordnik.com\"> http://swagger.wordnik.com</a> or on irc.freenode.net, #swagger. For this sample, you can use the api key \"special-key\" to test the authorization filters",
    "version": "1.0.0",
    "title": "Swagger Petstore API",
    "termsOfService": "http://helloreverb.com/terms/",
    "contact": {
      "name": "Wordnik API Team",
      "url": "http://developer.wordnik.com",
      "email": "apiteam@wordnik.com"
    },
    "license": {
      "name": "Apache 2.0",
      "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
    }
  }
```



Software AG Training | 11 - 14

Notes:

The screenshot shows the webMethods API Portal interface. At the top, there's a navigation bar with links for 'WEBSITES', 'API Portal', 'API Gallery', 'API Packages', 'App Gallery', and 'Support'. On the right side of the header, there are 'My account' and 'Logout' buttons. Below the header, the main content area has a breadcrumb trail: 'Home / API Gallery / BookingsAPI'. The main title is 'BookingsAPI'. To the left, there's a sidebar with sections like 'GETTING STARTED' (highlighted with a blue box), 'API DETAILS', 'API resources', 'API documents', 'Access API', and 'FURTHER INFORMATION' (with links to 'Latest posts' and 'Try API'). In the center, there's a large image of a cloud with the text 'BookingsAPI' and a small 'REST' icon. Below the image, there's a section titled 'About BookingsAPI' with a brief description and a note about booking a cruise. At the bottom of this section, there's a 'API RESOURCES' button. To the right, there's a sidebar with 'What's next?' options: 'Support forum', 'Export API as', and 'Download Client SDK'. Below that is a 'Rate this API.' button, which is also highlighted with a blue box. The status bar at the bottom right says 'Software AG Training | 11 - 15'.

Notes:

Software AG

Inspecting API using API Detail Page (2/3)

- API Resources
 - Description
 - Parameters
 - Request
 - Sample request
 - Sample response
 - Response
 - Details
 - HTTP status codes



```

{
  "customerID": "000001",
  "customerEmail": "Adam.Apple@email.com",
  "room": [
    {
      "roomID": "S05",
      "numRooms": "1"
    }
  ],
  "person": [
    {
      "emailAddress": "Adam.Apple@email.com",
      "title": "Mr",
      "firstname": "Adam",
      "surname": "Apple",
      "dob": "1992-09-01",
      "passportID": "1128143456789",
      "passportExpiration": "2022-07-01",
      "passmoCountry": "USA"
    }
  ]
}
  
```

Software AG Training | 11 - 16

Notes:

Inspecting API using API Detail Page (3/3)

- API documents
 - List of attached docs
 - API Policies
 - Enforced Policies
 - Access API
 - Endpoint Information
 - Endpoint URL

GETTING STARTED

[About BookingsAPI](#)

API DETAILS

[API resources](#)

[API documents](#)

[API policies](#)

[Access API](#)

FURTHER INFORMATION

[Latest posts](#)

[Try API](#)

About BookingsAPI

You found the perfect cruise, now lets get it booked. Before a cruise can be booked, the customer must sign-up to the SAGTours site use the Bookings API to book a cruise. You need the credentials you gave during sign-up to book the cruise. The credentials should be (the HTTP header Authorization: BASIC).

API resources

API documents

API policies

List of default policies.

Identify & Authorize Application

Require API Key Check

The API requires the usage of API Keys for Authentication and Authorization. There are 2 variants of passing along the API Key:

1. API Key as HTTP Header

Pass along the API key as x-Gateway-APIKey

Details

x-Gateway-APIKey:api-key-value

Example

x-Gateway-APIKey:jeffsfad-fsdaf-swerwe-fsdad

2. API Key as Query Param

Pass along the API key as URL Query Parameter - APIKey

Details

APIKey:u03dap/api-key-value

Example

<http://localhost:5555/v1/cus/Rest&API?APIKey=u03dap/api-key-value&sdaf-sd-fsdaf-eawouue-fsdad>

Access API

Latest posts

Try API

Software AG Training | 11 - 17

Notes:

Inspecting SOAP/WSDL based APIs

- SOAP/WSDL based APIs can be published
- Operations will get extracted
- Input/Output Message signatures will be autocreated from the WSDL

The screenshot shows the 'getYacht' API details page. On the left, there's a sidebar with links like 'About getYacht', 'API methods', 'API documents', and 'Assests API'. The main content area has a 'getYacht' logo and a 'About getYacht' section. Below it is the 'API methods' section, which is highlighted with a blue border. It contains a 'Sample usage' section with XML code for both 'Input message' and 'Output message'. The XML code is as follows:

```
<soap:Envelope>
  <soap:Header>
    <wsa:Action>http://schemas.xmlsoap.org/soap/encoding/</wsa:Action>
    <wsa:From>http://trainHost.webmethods.com:9400/samples/wsdlServices.YachtService.wsdl</wsa:From>
    <wsa:To>http://trainHost.webmethods.com:9400/samples/wsdlServices.YachtService.wsdl</wsa:To>
    <wsa:MessageID>urn:uuid:4a5a4a5a-4a5a-4a5a-a5a4-4a5a4a5a4a5a</wsa:MessageID>
  </soap:Header>
  <soap:Body>
    <ns1:getYachtFromDB>
      <ns1:main>http://trainHost.webmethods.com:9400/samples/wsdlServices.YachtService.wsdl</ns1:main>
      <ns1:yachts>
        <ns1:names>
          <ns1:name>Yacht 1</ns1:name>
          <ns1:name>Yacht 2</ns1:name>
        </ns1:names>
        <ns1:ownerIds>
          <ns1:ownerId>1</ns1:ownerId>
          <ns1:ownerId>2</ns1:ownerId>
        </ns1:ownerIds>
      </ns1:yachts>
    </ns1:getYachtFromDB>
  </soap:Body>
</soap:Envelope>
```

On the right side of the page, there are social sharing icons (Facebook, Google+, Email) and a 'Rate this API' section with a 'No ratings' message.

Notes:

Testing REST APIs in API-Portal

- API Portal allows to try out REST APIs
 - Select Sandboxes
 - Select Authentication schema
 - Choose REST resource
 - Supply parameters as required
 - Try out combinations of parameters and see how the API behaves
 - Introspect request and response headers

Software AG Training | 11 - 19

Notes:



Exercise 16

- Publishing API to API-Portal

Notes:



Lifecycle Control

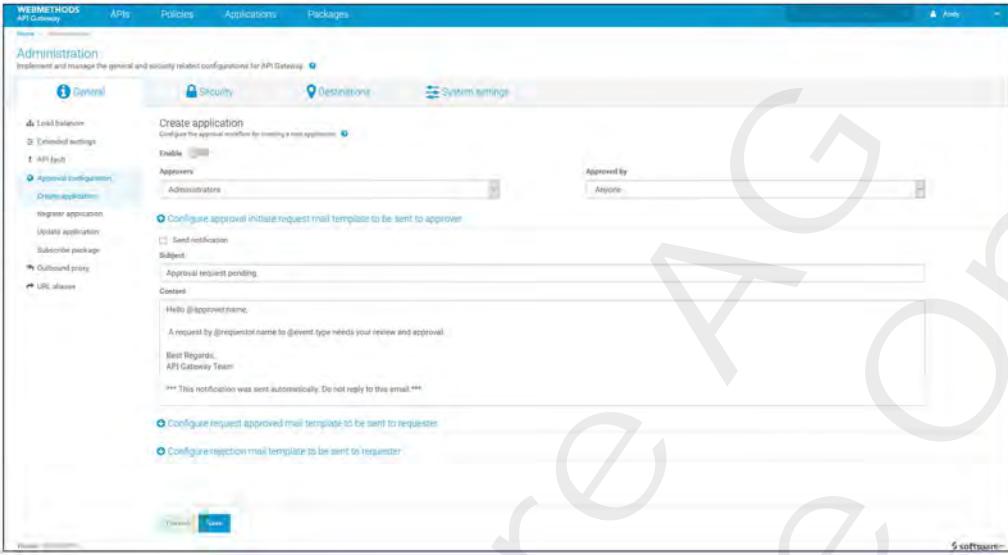
Notes:

API Grouping Information

- API Gallery ordering criteria is based on published metadata (**API-Portal Information** profile)

The screenshot shows the Software AG API Management interface. On the left, there's a modal window titled 'BookingAPI' for updating an API. It has tabs for 'Base Information', 'Technical Information', 'Discovery API endpoint', and 'API grouping'. The 'API grouping' tab is selected and highlighted with a red border. It contains a dropdown menu with options: 'None', 'BookingAPI', 'Customer Management', 'Commerce and Marketing', and 'Product Catalog'. The 'BookingAPI' option is also highlighted with a red border. On the right, the main interface shows the 'WEBSERVICE API-Portal' section with tabs for 'API Gallery', 'API Packages', 'App Gallery', and 'Swagger'. Below this, there are cards for 'SignupAPI' (with a note about BaaS), 'Customer Management', and 'BookingAPI' (with a note about BaaS). At the bottom, there's a footer with 'Software AG Training | 11 - 22'.

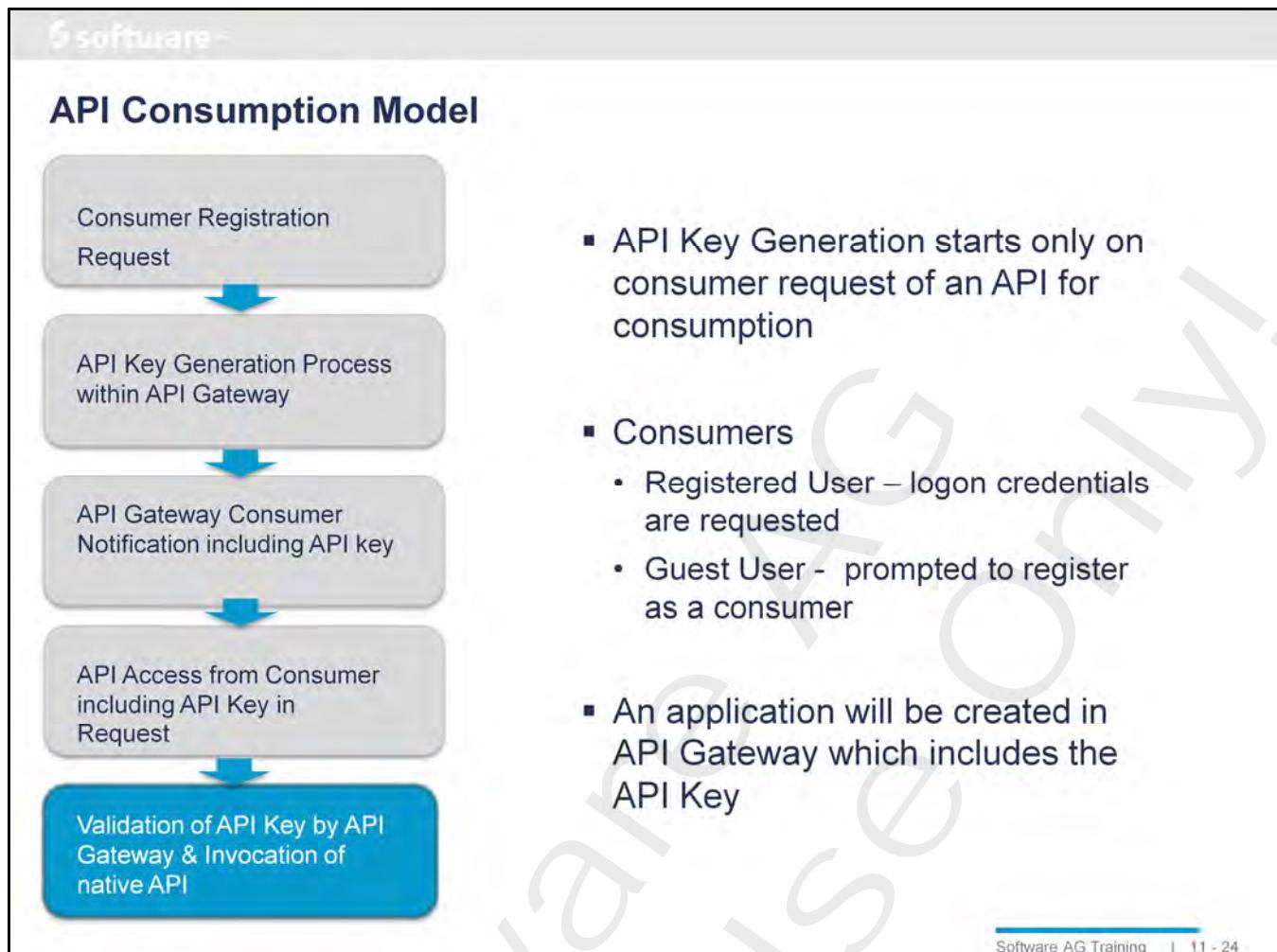
Notes:



The screenshot shows the 'Approval Workflow for Consumer Applications' configuration screen in the webMethods API Gateway. The left sidebar lists various administration options like Link balance, Extended settings, API Info, and Approval integrations. The main panel is titled 'Create application' and 'Configure the approval workflow by creating a new system'. It includes fields for 'Enable' (checkbox), 'Approvers' (dropdown set to 'Administrators'), and 'Approved by' (dropdown set to 'Anyone'). Below these are sections for 'Configure approval initiate request mail template to be sent to approver' (checkbox 'Send notification' checked, subject 'Approval request pending', content placeholder 'Hello @approver.name, A request by @requestor.name to @event.type needs your review and approval. Best Regards, API Gateway Team'), and 'Configure request approved mail template to be sent to requester' and 'Configure rejection mail template to be sent to requester' (both sections have a note '*** This notification was sent automatically. Do not reply to this email'). At the bottom are 'Forward' and 'Save' buttons.

Software AG Training | 11 - 23

Notes:



Notes:



Exercise 17

- Publishing SAGTours APIs to API-Portal

Notes:

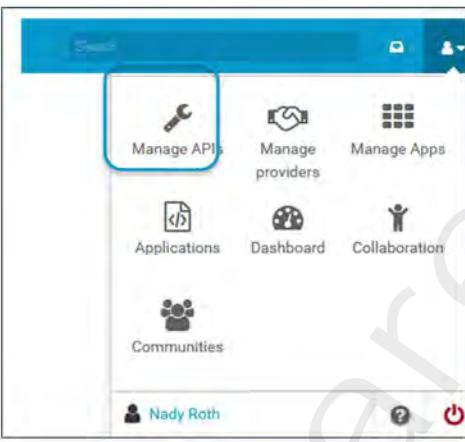


API Import in API Portal

Notes:

Direct API Import

- To address cases where customers intend to use API Portal without requiring CentraSite / API Gateway
 - Available to users in API-Provider role
 - Supported are RAML0.8, Swagger 2.0 and WSDL 1.x based APIs
 - Custom Importers can be added as well
- Only visible for users with permissions to add APIs



Software AG Training | 11 - 27

Notes:

Manage APIs

- Importing from File
- Import from URL
- Copy & Paste API Definition

The image contains three side-by-side screenshots of a software interface titled 'Import API'.
1. The first screenshot shows a 'Select File...' button and a 'Browse' button. Below it is a dropdown menu set to 'Swagger (2.0)' with a 'Import API' button at the bottom.
2. The second screenshot shows the same interface but with a dropdown menu containing three options: 'Swagger (2.0)', 'Swagger (0.8)', and 'WSDL (1.x)'.
3. The third screenshot shows the same interface with the 'Import API' button highlighted in blue.

- published APIs can be deleted here as well, but not updated

Software AG Training | 11 - 28

Notes:



12

API Portal as Consumer

Notes:

Objectives

At the end of this chapter you ...

- Know how to register to API Portal
- Know how to request an API-Key
- Know where to find the API-Keys within API Portal
- Know how to try out the API within API Portal using the API-Key

Notes:

Software AG Internal Use Only

API-Portal Landing Page

- Logon to API-Portal using URL as guest
 - <http://API-portal:18101/#default/home>

WEBMETHODS API Portal

API Gallery API Packages App Gallery Support

WEBMETHODS API Portal

End-to-end API management solution powered by Software AG

Log in Register

Develop using our APIs

View all APIs available in our API Gallery, explore their technical details, and try them.

Register for free

Learn how to join the user community and how to request access tokens for APIs.

Socialize

Visit our support forum to find out more about our APIs and to learn how other developers use them.

Software AG Training | 12 - 3

Notes:

Discovering APIs using API Gallery

- Dynamic rendering of API Gallery based on Metadata definitions
- Browse APIs based on configurable criteria
 - Maturity status (BETA APIs vs. Production APIs)
 - Business term (free vs. paid)
 - API group (Business Domains)



The screenshot shows the 'API Gallery' section of the webMethods API Platform. It displays two API entries:

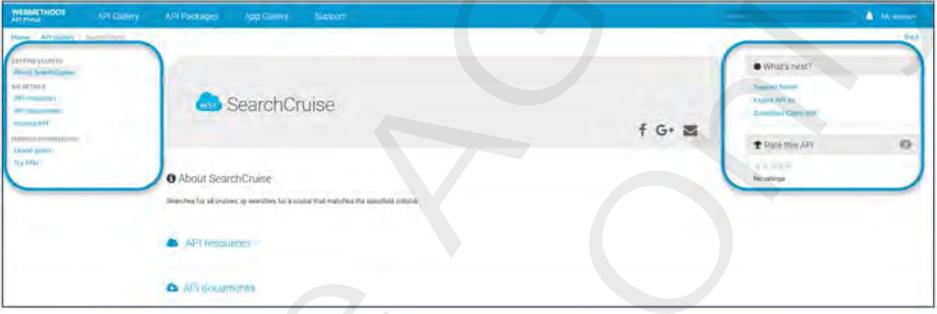
- SignupAPI**: REST API for signing up to SAGTours. It describes the process of creating a customer from SAGTours' Signup and Booking are protected via Basic Authentication. If an error occurs, the Sign-up API may give a text message as well as the HTTP response code.
- BookingsAPI**: REST API for booking a flight. It describes the process of booking a flight before a cruise can be finalized, and it also signs up to the SAGTours site if they have not already done so.

At the top right, there is a search bar and a 'Grouped by' dropdown menu. The bottom right corner contains the Software AG logo and the text 'Software AG Training | 12 - 4'.

Notes:

API Detail Definition

- Detailed API usage information
 - API Description
 - API Details (REST Resources and Methods, Semantics of Response Codes)
 - Usage Examples
- Try API>>
- What's next?
 - Support forum
 - Get access token
 - Export API
- Rate APIs

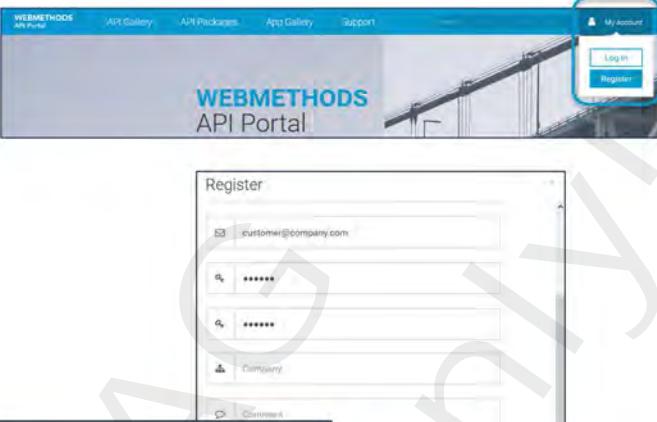
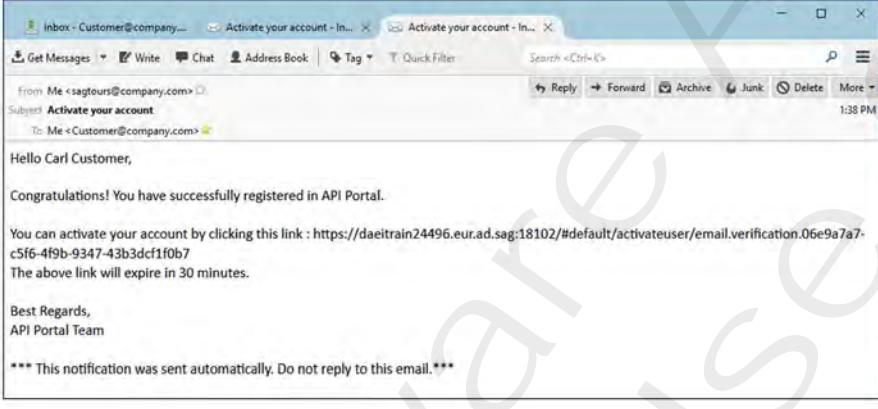


Software AG Training | 12 - 5

Notes:

Registering to API-Portal

- Registration request via
 - Register button in API-Portal Header
- API-Portal will
 - Add user to User Management
 - Assign user to API Consumer group
 - Send success eMail to requestor
 - Login the new user to API Portal

Software AG Training | 12 - 6

Notes:

Login to API-Portal

- User Account in API-Portal
 - Email / password
- User Menu available
 - Applications, Dashboard, Manage App, Collaboration,
 - Communities
- On API level
 - List of followers

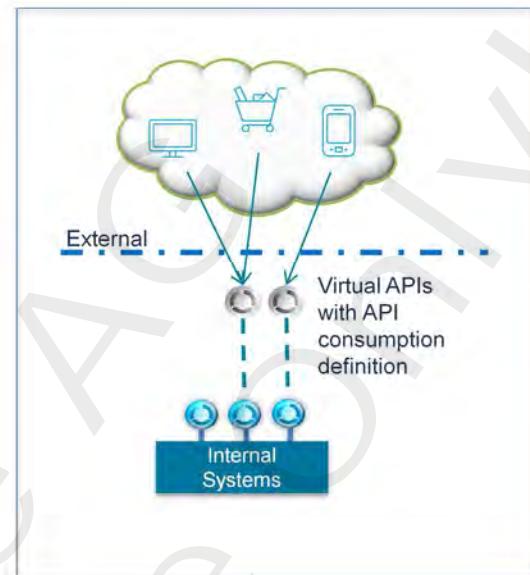
The screenshot shows the webMethods API Portal interface. At the top, there's a navigation bar with links for API Gallery, API Packages, App Gallery, and Support. Below the navigation, a sidebar on the left lists 'MY FAVORITE CRUISE' with options like 'About SearchCruise', 'API DETAILS', 'API resources', 'API documents', and 'Access API'. The main content area displays the 'SearchCruise' API details, including its logo (a blue cloud with a white cruise ship), social sharing buttons (Facebook, Google+, Email), and a search bar. To the right of the main content is a user menu with icons for Applications, Dashboard, Manage Apps, Collaboration, and Communities. Below the user menu, there are sections for 'Rate this API' (No ratings) and 'List of followers' (Empty list).

Software AG Training | 12 - 7

Notes:

Accessing APIs defined with Consumption Settings

- API Access Token needed
 - API Key
 - OAuth2
- API-Portal capabilities
 - Request API Keys and OAuth2
 - Try out and evaluate the API with personal access tokens
 - Manage access tokens/applications in the User menu



Software AG Training | 12 - 8

An Application Programming Interface Key is generated by CentraSite to identify the API, its provider or its consumer.

The API key acts as a unique identifier and a secret token for authentication. Generally it will have a set of access rights on the API.

Tightly integrated Workflow

- Application onboarding workflow
 - Users sign up in the API-Portal
- Access Token (API Key + OAuth2) provisioning workflows
 - User requests access token in the Portal
 - Provider approves access to API
 - Access tokens automatically provisioned to gateway infrastructure

The diagram illustrates the tightly integrated workflow. It starts with 'Consumers' at the top, indicated by a blue icon of two people. A large blue arrow points downwards from 'Consumers' to a screenshot of the 'WEBMETHODS API-Portal'. Below this screenshot, another blue arrow points downwards to a screenshot of the 'Welcome to webMethods API Gateway'. The entire process is enclosed in a light blue rectangular border.

Consumers

Discover & Signup

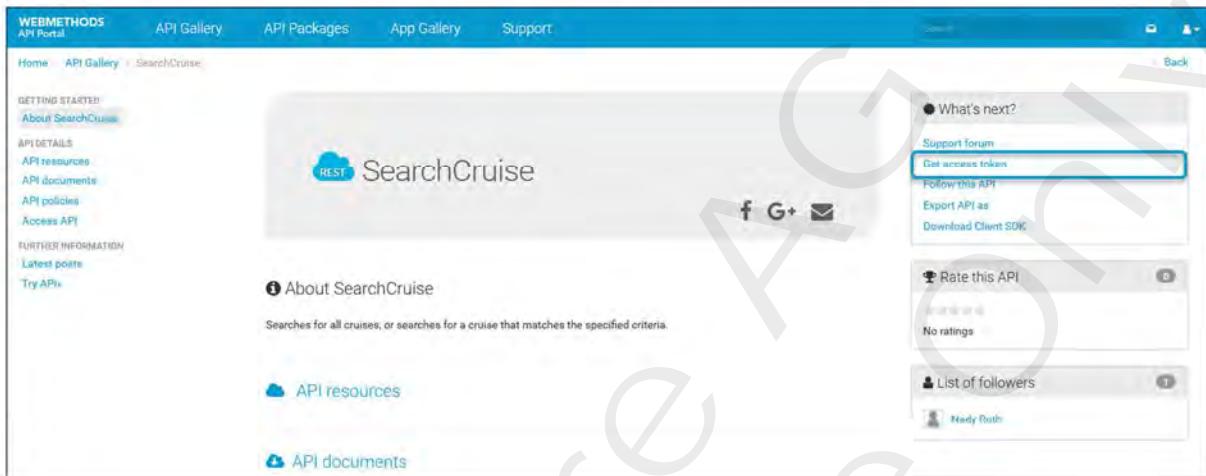
Create Application

Software AG Training | 12 - 9

Notes:

API Portal – Requesting Access Tokens

- Prerequisite is a registered user in API-Portal
- Request API Keys and OAuth2 access tokens for using the API



The screenshot shows the API Gallery page for the 'SearchCruise' API. The left sidebar has sections for 'GETTING STARTED', 'API DETAILS', and 'FURTHER INFORMATION'. The main content area displays the 'SearchCruise' logo and a brief description: 'Searches for all cruises, or searches for a cruise that matches the specified criteria.' Below this are links for 'API resources', 'API documents', and 'About SearchCruise'. On the right side, there are sections for 'What's next?', 'Support forum' (with a highlighted 'Get access token' link), 'Rate this API' (with a note 'No ratings'), and 'List of followers' (with a note 'No followers'). A watermark 'Internal Use Only' is diagonally across the page.

Software AG Training | 12 - 10

Notes:

Configuring the Request for an Application

- API Access token will be
 - delivered via eMail
 - Available in API-Portal
- User will be created in the organization defined in API-Portal Definition
- API Access token will be added as consumer to the API in API Gateway
- Try out and evaluate the API with personal access tokens
- Manage access tokens in user preferences page

Request API access token

Please provide the required information below to receive a personalized access token via e-mail.

Application name

Application description

Application redirect URI

Software AG Training | 12 - 11

Notes:

The screenshot displays two overlapping pages from the webMethods API Portal. The top page shows a user profile for 'Carl Customer' with a blue user icon. The left sidebar menu has a section for 'APPLICATIONS' with a sub-item 'My applications' highlighted by a red box. The bottom page shows a detailed view of a 'Customer Application'. This view includes a 'Basic information' section with a description: 'Customer portal to offer worldwide e-business.' and a purpose: 'API to describe portal'. Below this is an 'Access tokens' section containing an API key ('pm1589bf-4598-4b23-9c8c-b2157abb5d68') and an API access key ('Unlimited'). A large watermark reading 'Internal Software AG Use Only!' is diagonally across the page.

Software AG Training | 12 - 12

Notes:

The screenshot shows the webMethods API Gateway interface. At the top, there are tabs for 'WEBMETHODS API Gateway', 'APIs', 'Policies', 'Applications', and 'Packages'. The 'Applications' tab is highlighted with a blue border. Below the tabs, there's a search bar labeled 'SearchCruise' and a section titled 'Selected applications'. A blue arrow points from the left margin to the 'Customer Application' row, which contains three items: 'MyFirstApplication', 'MyApplicationJustAPKey', and 'Customer Application'. The 'Customer Application' item is selected, as indicated by a blue border around its row. A modal window titled 'Customer Application' is open, displaying detailed information about the application, including its access key, client ID, client secret, and scopes.

Software AG Training | 12 - 13

Notes:

The screenshot shows the webMethods API Portal interface. At the top, there's a navigation bar with links for 'API Portal', 'API Gallery', 'API Packages', 'App Gallery', and 'Support'. Below the navigation, a header says 'Try API Using API-Key'.

The main area displays an API endpoint for 'SearchCruise'. The URL is listed as `GET http://daetrain26233.5555/gateway/SearchCruise/1.0/cruises`. There are 'Clear' and 'Test' buttons next to it.

On the left, there's a sidebar with sections for 'APPLICATION' (set to 'Customer Application'), 'Basic Services' (listing 'GET /cruises' and 'GET /cruisesWithCruiseID'), and 'Description' (which includes details about 'endDate' and 'startDate').

The main configuration area has two sections: 'Query parameters' and 'Header parameters'. Under 'Query parameters', there are fields for 'Name' (e.g., 'endDate', 'startDate') and 'Value'. Under 'Header parameters', there are tabs for 'Headers' (selected) and 'Security'. In the 'Headers' tab, there's a table with columns 'Name' and 'Value'. One row is highlighted with a blue border, showing 'x-Gateway-APIKey' with the value 'as1569bf-4598-4b23-9e0c-021571'. There are also '+' and '-' buttons for adding or removing rows.

At the bottom right of the main area, there's a footer with the text 'Software AG Training | 12 - 14'.

Notes:



Exercise 18

- Using API Portal as Consumer

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



13

Analytics in API Portal

Notes:

Objectives

At the end of this chapter you ...

- Know
 - How to use Dashboards in API Portal
 - How to define API Portal as Event Destination in API Gateway
 - How to configure Policies in API Gateway to send events to API Portal

Notes:

API Portal Dashboarding

- Integrated usage dashboard
 - Global Dashboard
 - API audit log
 - Runtime dashboard
 - Runtime dashboard
 - API trends dashboard
 - Consumer dashboard
 - Support of
 - Setting filters interactively to drill down to different levels of information



Software AG Training | 13 - 3

Notes:

Global Dashboard

- Page views
- Page views Guest Vs. Registered
- Page views over time
- Visits per API
- API views over time

The dashboard displays several data points:

- A bar chart titled "Page views" showing the number of page views for different project categories. The data is as follows:

Project Category	Number of Page Views
Project A	15
Project B	10
Project C	8
Project D	6
Project E	4
Project F	3
Project G	2
Project H	1

- A pie chart titled "Page views Guest Vs Registered" showing the distribution between guest and registered users.
- A chart titled "Visits per API" showing the count of visits for the top 10 APIs. The data is as follows:

API	Visits
API A	10
API B	8
API C	6
API D	4
API E	3
API F	2
API G	1

- A chart titled "API views over time" showing the number of views for the top 10 APIs over time.

Notes:

Runtime Dashboard

- Overall Events: Pie chart is sliced based on event types to have a better understanding about the overall behaviour of all the API's hosted.
- Top APIs by consumption : This chart shows the APIs that are most frequently accessed by consumers.
- Top 10 APIs: to identify top consumer applications through which the APIs are accessed frequently.
- Runtime Events : overview about the events that are captured in the system through which preliminary issue analysis could be done.
- Requests per user
- Requests over time by API

Time	UserAgent	Method	RequestID
12:42:36.122	Firefox/3.6.12	GET	12:42:36.122
12:42:36.117	Firefox/3.6.12	PUT	12:42:36.117
12:42:36.089	Firefox/3.6.12	PUT	12:42:36.089

Software AG Training | 13 - 5 -

Notes:

Pre-requisite for API Runtime Event Dashboard Information

- API Portal Events configuration in API Gateway
- API Portal destination is available when you configure a policy on the API in API Gateway

The screenshot shows the WEBMETHODS API Gateway Administration interface. On the left, there's a sidebar with various options like API Gateway, API Portal, Configuration, Events, Audit Log, CentraSite, Database, Digital Events, Elasticsearch, and Configuration. The main panel has tabs for General, Security, Destinations, and System settings. Under Destinations, it shows 'Event types' (Error, Lifecycle, Policy violation) and 'Performance metrics data' (Report performance data, Publish interval: 2 minutes). A modal window titled 'Policy properties' is open, showing 'Log Invocation' options (Store Request Payload, Store Response Payload, Compress Payload Data, Log Generation Frequency: Always) and a 'Destination *' section. The 'Destination *' section contains two checked checkboxes: 'API Gateway' and 'API Portal'. Other destination options listed include Audit Log, CentraSite, Digital Events, Elasticsearch, Email, JDBC, Local Log, and SNMP. The modal has an 'Open in full-screen' button at the top right.

Notes:

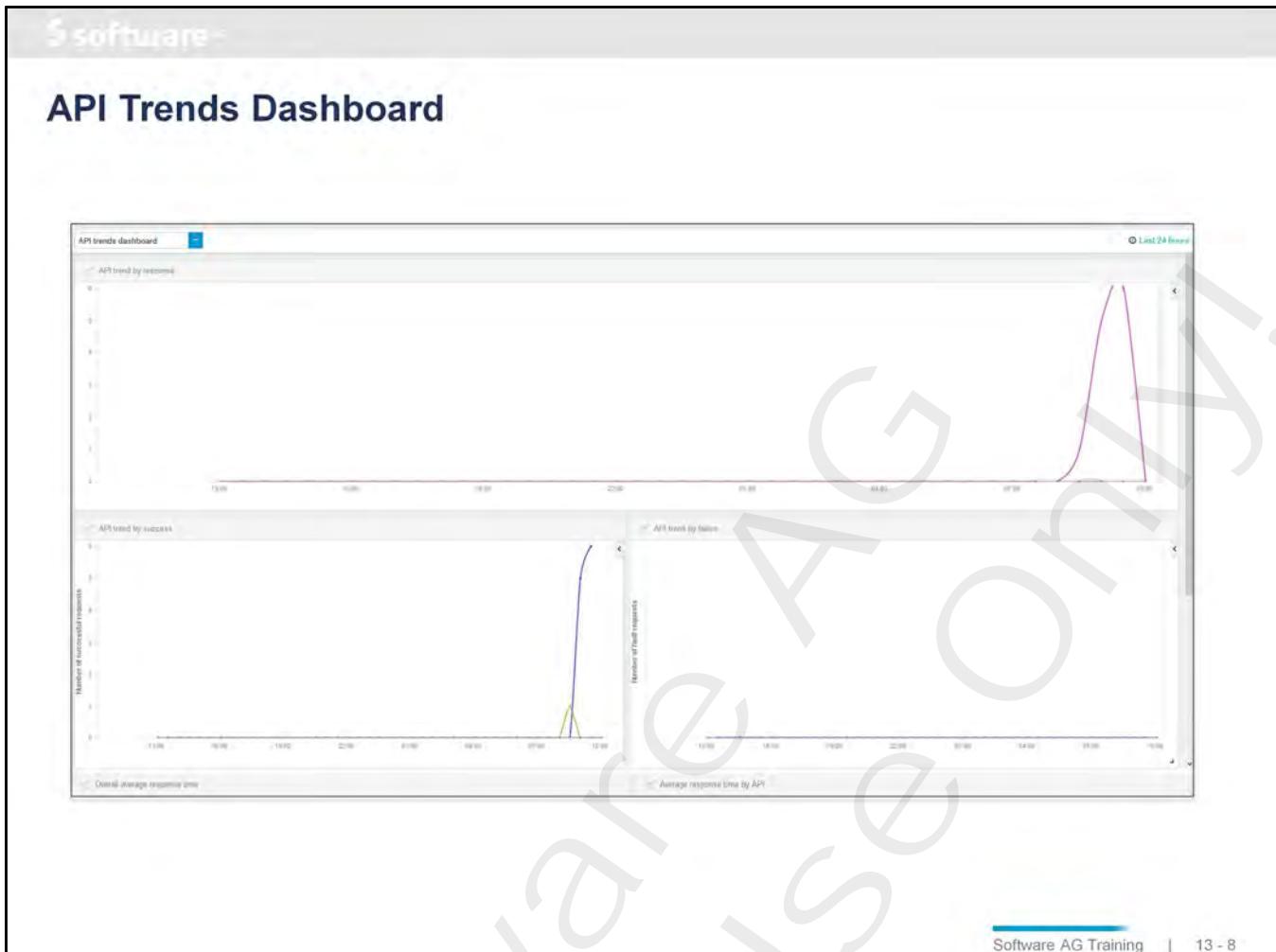
API Portal Destination for Policies

- API-Portal destination is available for these Policy Actions
 - Log Invocation
 - Monitor Service Performance
 - Monitor Service Level Agreement
 - Throttling Traffic Optimization

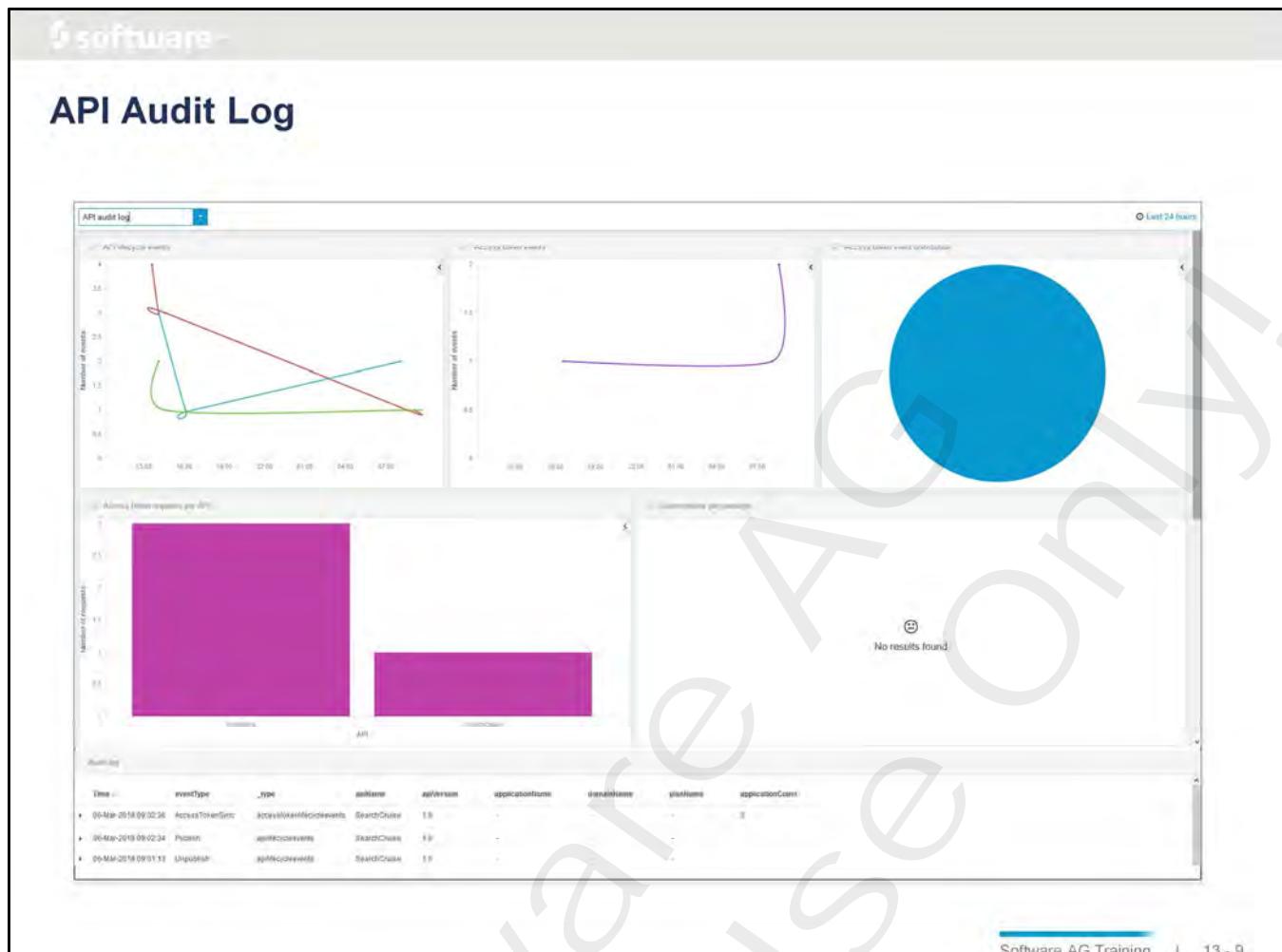
The screenshot displays three separate configuration panels side-by-side:

- Log Invocation:** Contains checkboxes for "Store Request Payload" and "Store Response Payload". A dropdown menu for "Log Generation Frequency*" shows "Always". Below is a "Destination" section with checkboxes for API Gateway, API Portal, and CentralSite.
- Monitor Service Performance:** Contains an "Action Configuration" section with a "Add action configuration" button. It includes a "Destination" section with checkboxes for API Gateway, API Portal, and CentralSite, followed by sections for "Alert Interval" (set to 1 minute), "Alert Frequency" (set to "Only Once"), and "Alert Message" (set to "Average response time for consumer exceeded").
- Monitor Service Level Agreement:** Contains an "Action Configuration" section with a "Add action configuration" button. It includes a "Destination" section with checkboxes for API Gateway, API Portal, and CentralSite, followed by sections for "Alert Interval" (set to 1 minute), "Unit" (set to Minutes), "Alert Frequency" (set to "Only Once"), and "Alert Message" (set to "Quota reached").

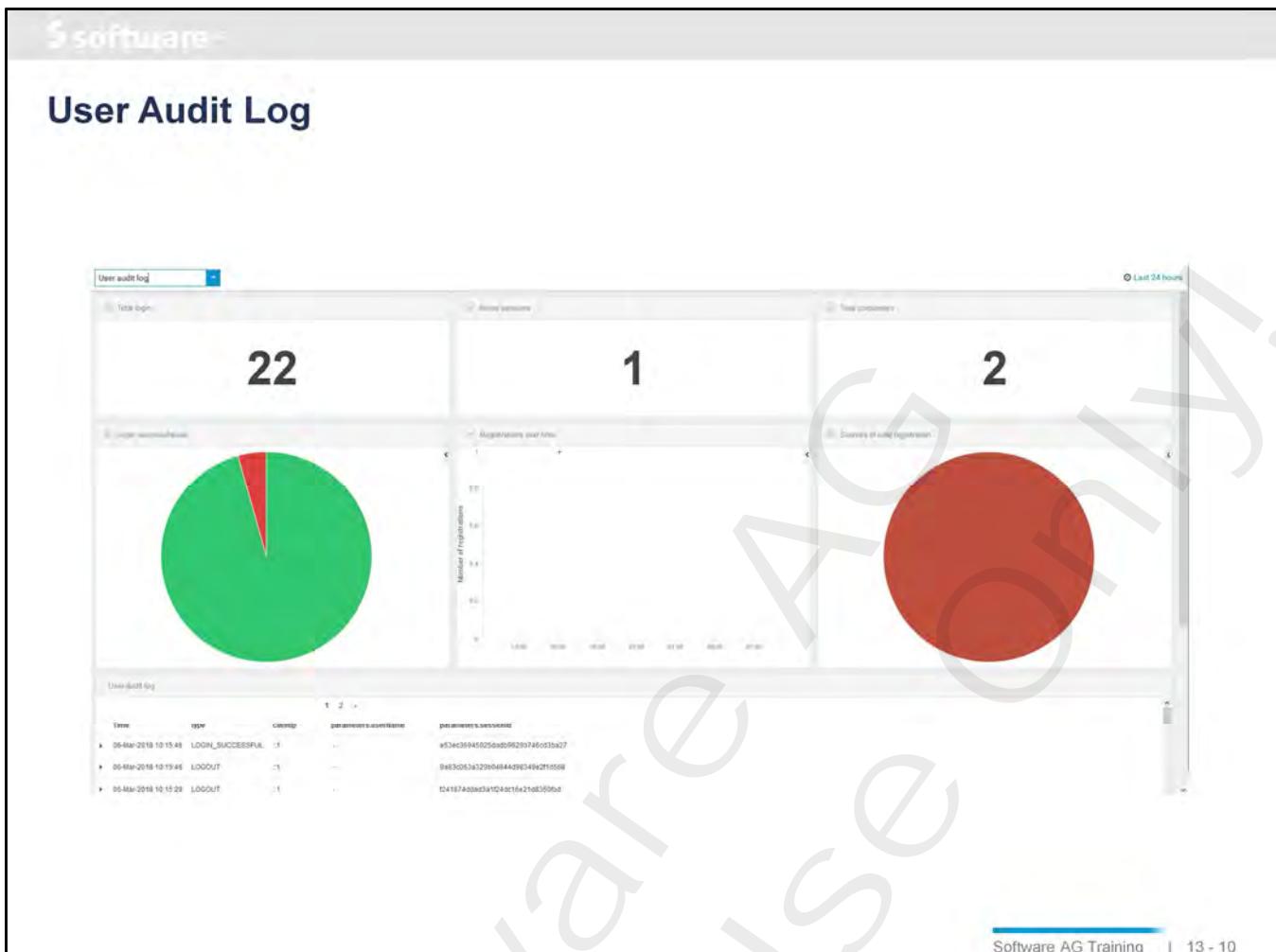
Notes:



Notes:



Notes:



Software AG Training | 13 - 10

Notes:

Consumer Dashboard

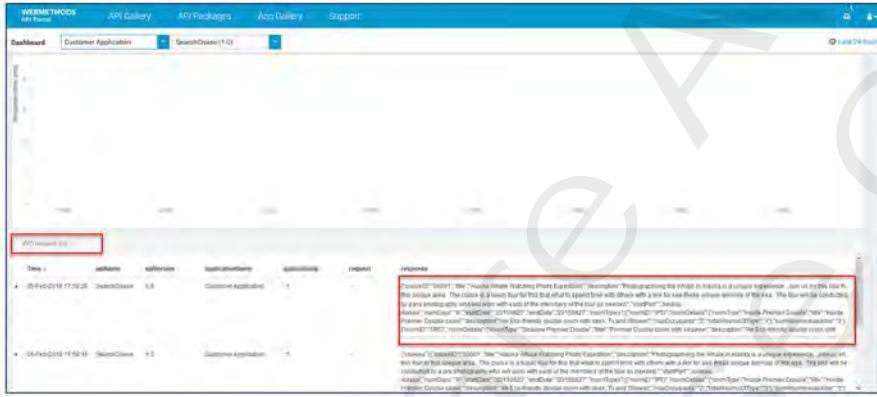
- Total requests (for different time intervals)
- Requests per API
- Requests over time by API
- Average response time
- Request Log

Software AG Training | 13 - 11

Notes:

Consumer Dashboard Filter

- Filter options
 - By application
 - By API
 - By Time
 - API request log



Software AG Training | 13 - 12

Notes:



Exercise 19

- Send Events to API Portal

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



14

Packages and Plans

Notes:

Objectives

At the end of this chapter you ...

- Know how to monetize your APIs by API Packages and Plans
- Understand the combination of APIs to Packages and Plans
- Can define a Plan and a Package associated with APIs in API Gateway
- Can publish a package to API Portal as API Provider
- Can subscribe to a Plan within a Package and consume the corresponding APIs

Notes:

Chapter Contents

- Monetization Overview
- Elastic Search REST API for Monetization

Notes:



Monetization Overview

Notes:

Overview

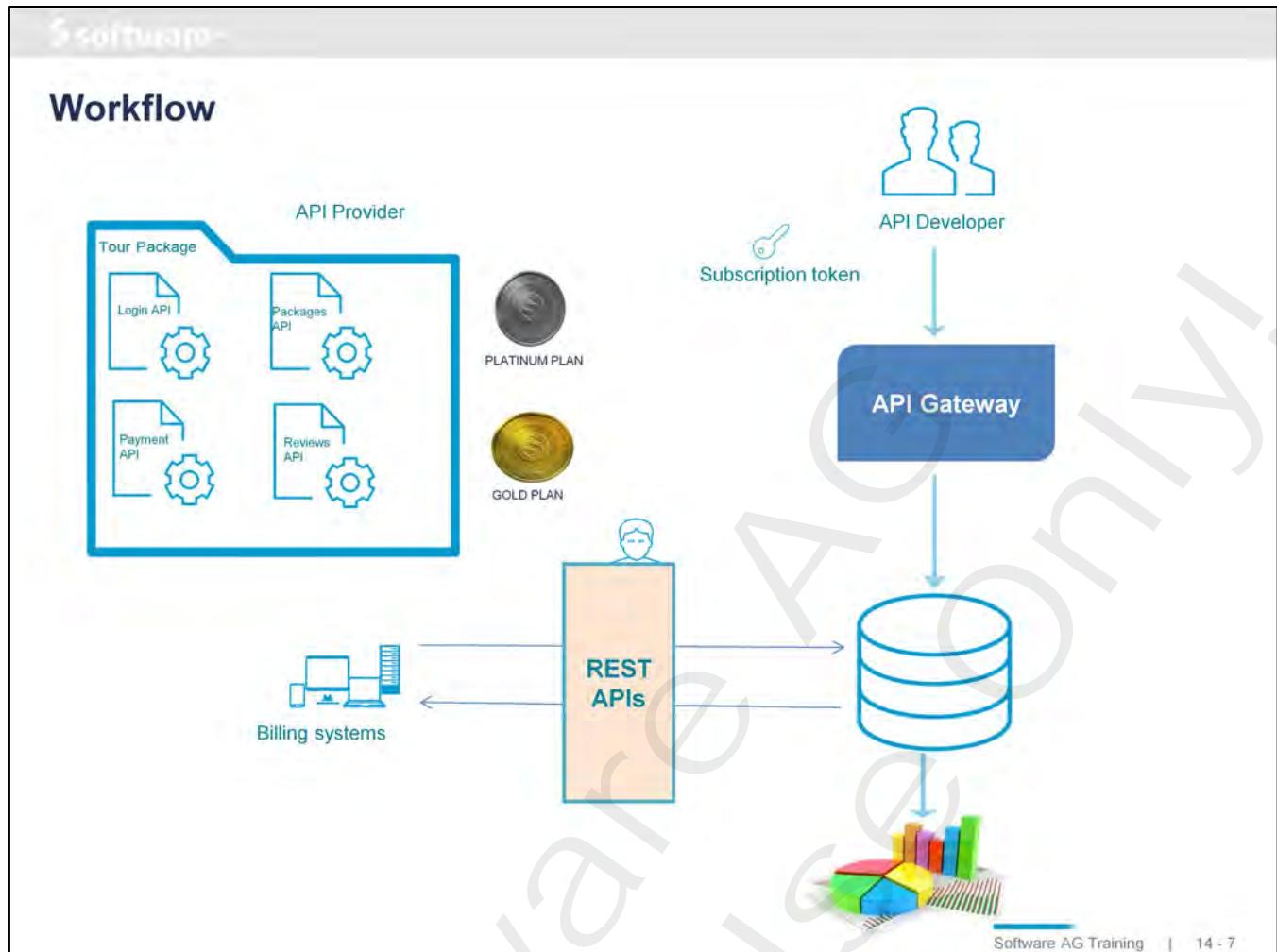
- API Provider exposes APIs to external Developers
 - The developers will build their own applications using these APIs
 - These applications will be consumed by end users
- Providing APIs to external Users
 - For free
 - generate revenue through various business models chosen by the API Provider
 - Quality of Service must somehow be guaranteed / monitored
- Monitoring & measuring the performance of the API
 - To know the success of an API or to claim that the API is profitable
 - Identify better avenues for revenue generation using existing data
 - Understand the consumers' interests and introducing attractive plans and services

Notes:

Monetization in API Gateway

- APIs are bundled as Packages and are associated with Plans
- Consumers can subscribe to the Package-Plan combination from API Portal
- Consumers access the APIs with their applications subscription key
- A transaction logging global system policy generates events for all API invocations
- API Providers can visualize the performance and invocations of the packages and applications through the dashboards available in API Gateway

Notes:



Notes:

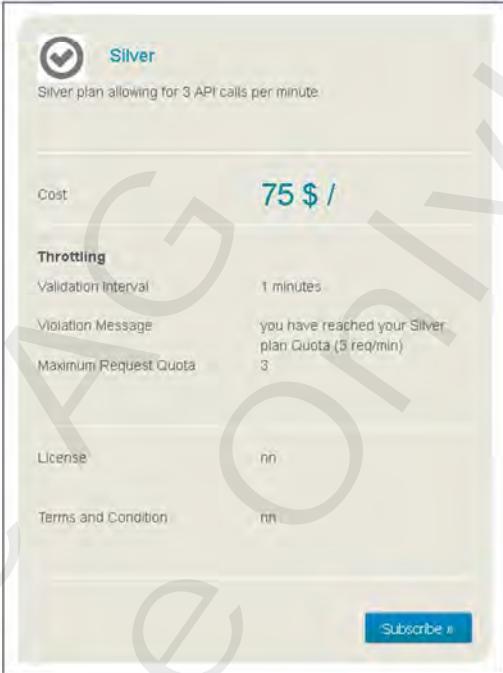
Packages as Exposure of APIs

- API Package
- Logical grouping of multiple APIs from a single API provider
 - 1 Package can contain 1 to n APIs
 - 1 API can belong to 1 to n Packages
- API Plan
- Contract proposal presented to consumers
- Offering
 - Varying availability guarantees
 - SLAs
 - Cost structures
- API Package
- Can be associated with multiple plans

Notes:

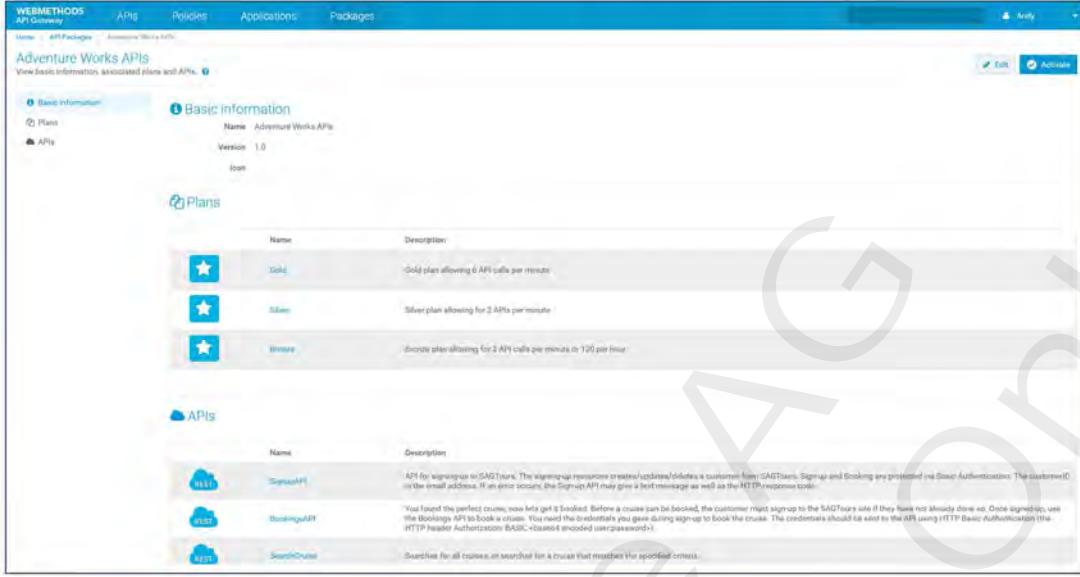
API Plans

- Expectation => Monetization Support
- Expose to consumers
 - Charge back services
 - Costs
 - 75 \$ / Month
 - ...
 - Throttling
 - Maximum request quota
 - Violation interval
 - Violation message
 - License
 - Terms
 - Plan subscriptions
 - Easy onboarding of consumers



Software AG Training | 14 - 9

Notes:

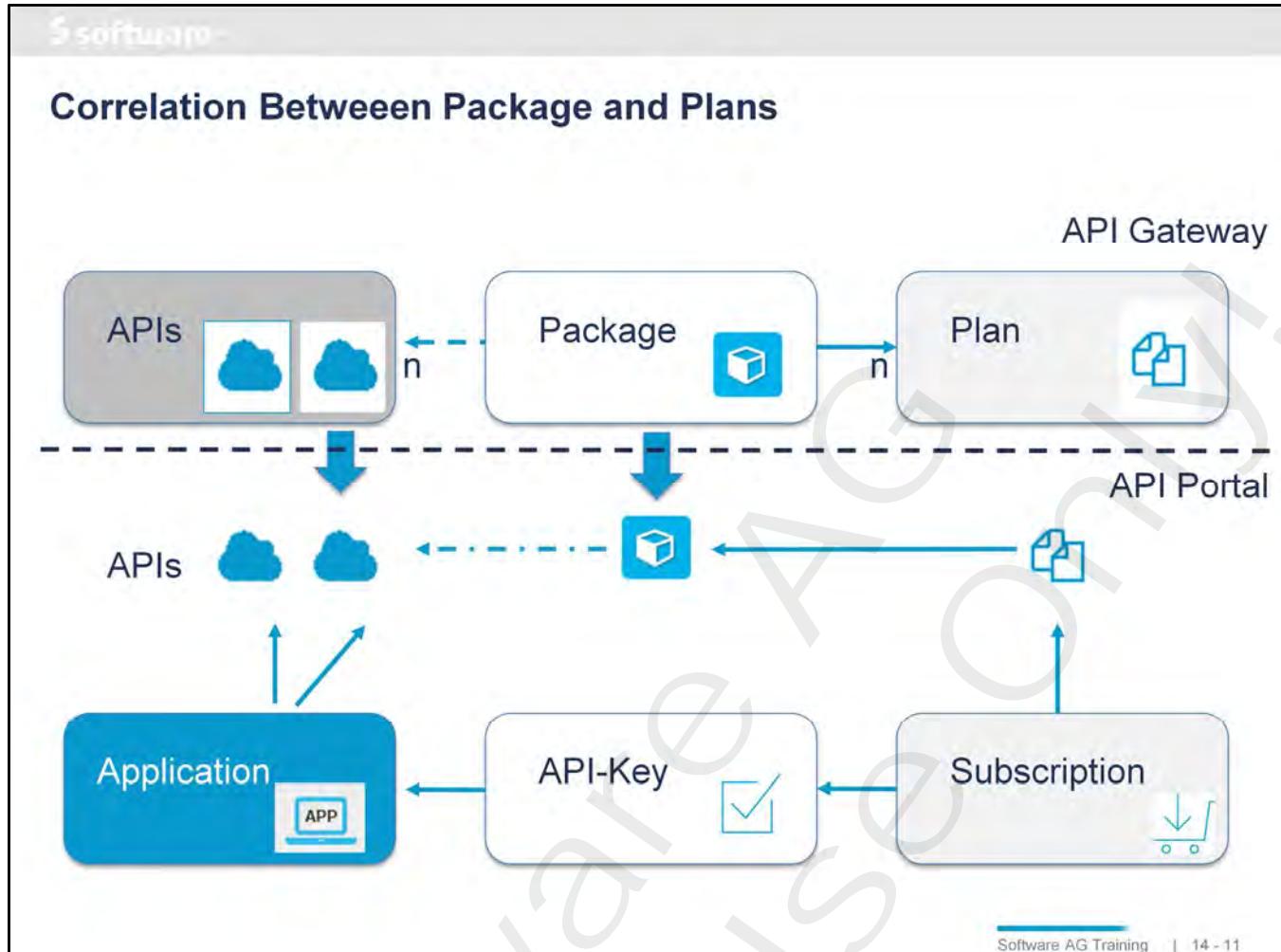


The screenshot shows the WEBMETHODS API Gateway interface. The top navigation bar includes links for Home, API Packages, Policies, Applications, and Packages. The main content area is titled "API Packages and Plans for Monetization". Under "Basic Information", it shows the package name "Adventure Works APIs" and version "1.0". There are tabs for "Plans" and "APIs". The "Plans" section lists three plans: "Gold" (allowing 6 API calls per minute), "Silver" (allowing 2 API calls per minute), and "Bronze" (allowing 1 API call per minute or 120 per hour). The "APIs" section lists three APIs: "SignupsAPI", "BookingsAPI", and "SearchCrusoe". A large watermark reading "Internal Use Only!" is diagonally across the page.

- API monetization
 - Create API packages and API service plans
 - Publish them to API Portal and manage subscriptions

Software AG Training | 14 - 10

Notes:



Notes:

Publishing API Packages to API Portal

- Publishing API Packages requires
 - Package to be activated
 - Included APIs to be already published to API Portal
- Publishing of API Packages to API-Portal
 - also publishes all referenced Plans



The screenshot shows the WEBMETHODS API Gateway interface. In the top navigation bar, there are tabs for APIs, Policies, Applications, and Packages. The Packages tab is selected. Below the navigation, there's a section titled "Manage packages and plans" with a sub-section "Create and manage packages and plans". There are two buttons: "Packages" and "Plans". The "Packages" button is highlighted. Below these buttons is a table with columns: Name, Description, Version, APIs, and Plans. A single row is visible for "Adventure Works APIs", which has a version of 1.0, 3 APIs, and 3 Plans. To the right of the table are several icons: a gear icon (highlighted with a blue box and arrow), a cloud icon (highlighted with a blue box and arrow), a user icon, and a trash bin icon.

Software AG Training | 14 - 12

Notes:

The screenshot shows the WEBMETHODS API Portal interface. At the top, there are tabs for API Gallery, API Packages, App Gallery, and Support. The main content area is titled "WEBMETHODS API Portal" and displays three API packages: BookingAPI, SearchCruise, and SignupAPI. Below this, there is a section titled "Plans" showing three subscription plans: Bronze, Free, and Silver. Each plan has its cost (50 \$, 0 \$, 75 \$), throttling limits (Maximum Request Quota, Validation Interval, Validation Message), and license terms. The "Subscribe" button is highlighted with a blue box for the Free plan.

Software AG Training | 14 - 13

Notes:

Subscribing to a Package

- Requires having a user account in API-Portal
- Sends a registration request to API Gateway
 - Provide Application Name and Description
- API Gateway creates the consumer Application and generates an
 - Access Key or OAuth Token
 - Embedded in the application
- API Gateway publishes Application and API access token back to the requestor on API-Portal
- API consumer can use API-Portal Try Out option to verify API consumption based on Plan limits (Throttling)

Notes:

The screenshot shows the webMethods API Portal interface. The main title is "Subscriptions in API Portal". Below it, there's a navigation bar with links like "API Gallery", "API Packages", "App Gallery", and "Support". The main content area displays a list of subscriptions, with one specific subscription highlighted. This highlighted subscription is shown in a detailed view. The detailed view includes:

- Name:** Custom SAOTours App/Subscription
- Description:** SearchCruise
- URL:** GET http://dastran/26233:5550/gateway/SearchCruise/1.0/cruises
- Query parameters:** None listed.
- Header parameters:**
 - Headers:** Accept, Content-Type
 - Content-Type:** x-Gateway-APIKey: cb3db057-ba32-4780-a66b-21209

Software AG Training | 14 - 15

Notes:

Miscellaneous

- Package Expansion
 - Even after a Package is published you can add more APIs to be part of the package
 - Existing / new subscribers will be able to access the newly created APIs with their existing subscription keys
- Package Shrinking
 - Even after a Package is published you can remove one or more APIs belonging to a Package
 - Existing / new subscribers will not be able to access the removed APIs with their subscription keys

Notes:



Elastic Search REST API for Monetization

Notes:

Elastic Search Data Requests

Scenario	Methods	Endpoint	Comments
GET all APIs	GET	http://localhost:9240/gateway/default/apis/_search	
Get all Policies	GET	http://localhost:9240/gateway/default/policies/_search	
Get all Applications	GET	http://localhost:9240/gateway/default/applications/_search	
Get Events count	GET	http://localhost:9240/gateway/default_analytics/transactionalEvents/_count	
Get Events List	GET	http://localhost:9240/gateway/default_analytics/transactionalEvents/_search	
Ping	Get	http://localhost:9240/	<pre>{ "name": "VMMEDCLUST01W_eur.ad.sag1484822599907", "cluster_name": "SAG_EventDataStore", "version": { "number": "2.3.2", "build_hash": "b9e4a8ac4d008027e4038f6abed77fdb4346f94", "build_timestamp": "2016-04-21T16:03:47Z", "build_snapshot": false, "license_version": "5.5.0" }, "tagline": "You Know, for Search" }</pre>

Software AG Training | 14 - 18

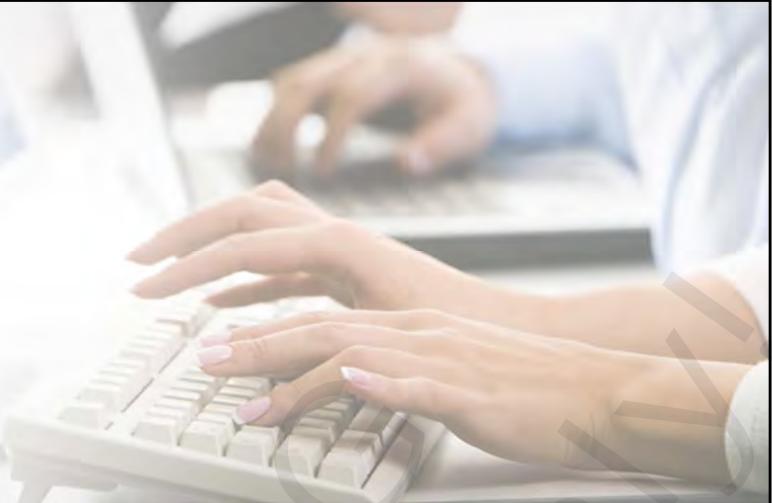
Notes:

Rest API to Extract Monetization Data

Details	URL
<p>Count of invocations for an API, Package, Plan or Application</p> <p>Output:</p> <pre data-bbox="317 462 495 512">{ "count": [{ "apiName": "PetStoreTest", "apiVersion": "1.0.0", "count": 2 }] }</pre>	<p><a href="http://<Server>:<port>/rest/apigateway/transctionalEvents/_count?apiName=PetStoreTest&httpMethod=post&from=2017-02-16">http://<Server>:<port>/rest/apigateway/transctionalEvents/_count?apiName=PetStoreTest&httpMethod=post&from=2017-02-16</p>
<p>Data of the invocation for an API,</p> <p>Output:</p> <pre data-bbox="317 539 495 642">{ "transaction": { "creationDate": "1487229288461", "apiName": "PetStoreTest", "apiVersion": "1.0.0", "apiId": "73c40484-678a-4f0d-a58c-2111869e5f4f", "totalTime": 1381, "responseCode": 1336, "operationName": "testApp", "applicationId": "10727b01-e13a-4c5d-b39e=c5c338101e21", "status": "SUCCESS", "totalDataSize": 1341, "responseCode": 200, "operationName": "/pet", "httpMethod": "post", "packageName": "TestPackage", "packageId": "7aaef5c8-3ea4-454f-a718-d9ff3288672a", "planName": "TestPlan", "planId": "70369925-2719-4c8a-b078-ca057516735e", ... }}</pre>	<p><a href="http://<Server>:<port>/rest/apigateway/transctionalEvents/_search?apiName=PetStoreTest&httpMethod=post&from=2017-02-16">http://<Server>:<port>/rest/apigateway/transctionalEvents/_search?apiName=PetStoreTest&httpMethod=post&from=2017-02-16</p>

Software AG Training | 14 - 19

Notes:



Exercise 20

- Managing API Packages and Plans

Notes:



15

Advanced Security

Notes:

Objectives

At the end of this chapter you ...

- Know about specific security configurations
 - Kerberos
 - SAML
 - JWT
 - OpenID
 - OAuth2
 - Encryption
 - Signing
 - Timestamp
 - ...

Notes:

Chapter Contents

- Overview
- Authentication on Transport Layer
- Kerberos Authentication in Message
- SAML Authentication/Authorization in Message
- Message Encryption/Signing
- OAuth 2 Support

Notes:



Overview

Notes:

Inbound Authentication – Transport Layer

- Supported for REST APIs and SOAP APIs
- API Provider can use API Gateway for enforce authentication on the API
- API Gateway policies can be configured so that API Gateway expects the clients to pass the authentication credentials through transport headers
- Supported Authentication Schemes
 - Basic Authentication
 - Kerberos Token Authentication
 - OpenID Authenticaiton
 - JWT Authentication

Inbound Authentication - Transport

Kerberos Token Authentication

Service Principal Name*

Service Principal Password*

HTTP Basic Authentication

OpenID Authentication

JWT Authentication

Software AG Training | 15 - 5

Notes:

Outbound Authentication – Transport Layer

- Supported for REST APIs and SOAP APIs
- The native API is protected and expects the authentication credentials to be passed through transport headers
- API Gateway Policies can be used to provide the credentials that will be added to the request and send to the native API
- Supported Authentication Schemes
 - Basic Authentication
 - Kerberos
 - NTLM
 - OAuth2
 - JWT
 - Alias

Outbound Authentication - Transport

Authentication scheme

Basic
Kerberos
NTLM
OAuth2
JWT
Anonymous
Alias

Username*

Password*

Domain

Software AG Training | 15 - 6

Notes:

Outbound Authentication - Transport - Properties

- Authentication Scheme
 - as seen on previous slide
- Authentication Mode, as subset based on AuthenticationScheme
 - Custom Credentials
 - Delegate incoming credentials
 - Incoming HTTP Basic Authentication
 - Incoming OAuth token
 - Incoming JWT
 - Transparent
- Additional properties based on mode
 - Custom Credentials
 - Username, Password, Domain
 - Client Principal, client Password, ...
 - OAuth2 token
 - ...

Software AG Training | 15 - 7

Notes:

Inbound Authentication - Message Layer

- Only supported for SOAP APIs
- API Provider can use API Gateway to enforce authentication on the API
- API Gateway policies can be configured so that API Gateway expects the clients to pass the authentication credentials through the payload message
- Supported Authentication Schemes
 - X.509 Certificate
 - WSS Username
 - SAML
 - Kerberos
 - Signing
 - Encryption

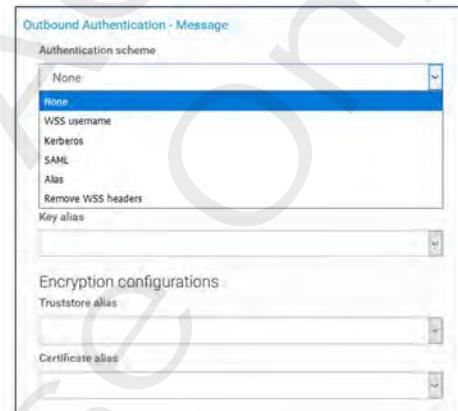


Software AG Training | 15 - 8

Notes:

Outbound Authentication - Message Layer

- Only supported for SOAP APIs
- The native API is protected and expects the authentication credentials to be passed through transport headers
- API Gateway Policies can be used to provide the credentials that will be added to the payload of the request and sent to the native API
- Supported Authentication Schemes
 - WSS Username
 - SAML
 - Kerberos
 - Alias
 - Remove WSS headers



Software AG Training | 15 - 9

Notes:

Outbound Authentication - Message - Properties

- Authentication Scheme
 - As seen on previous slide
- Authentication Mode, subset based on AuthenticationScheme
 - Custom Credentials
 - Delegate incoming credentials
 - Incoming HTTP Basic Authentication
- Additional properties based on scheme
 - Custom Credentials
 - Username, Password, Domain
 - Client Principal, client Password, ...
- Signing configurations
- Encryption configurations

Software AG Training | 15 - 10

Notes:

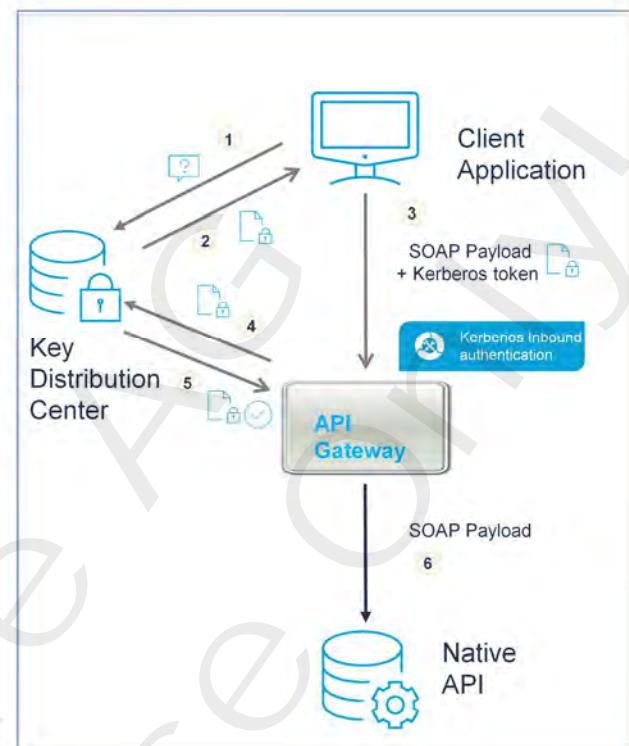


Authentication on Transport Layer

Notes:

Kerberos Inbound Authentication - Transport

1. Client sends requests for Kerberos token to Key Distribution Center (KDC)
 2. KDC sends Kerberos token to client
 3. Client adds the Kerberos token to the request and sends to the API Gateway
 4. API Gateway sends Kerberos Validation request to KDC
 5. KDC send Kerberos Validation to API Gateway
 6. API Gateway sends request to native service



Software AG Training | 15 - 12

Notes:

Kerberos Inbound Transport - Policy Configuration UI

- API providers can enforce transport level Kerberos authentication for an API by specifying either the User or Host-based SPN at the inbound

Notes:

Kerberos Outbound Authentication - Transport

1. Client sends requests to API Gateway
2. API Gateway sends Kerberos token request to Key Distribution Center (KDC)
3. KDC send Kerberos Token to API Gateway
4. API Gateway sends request and embedded Kerberos token to native service

Client Application

API Gateway

Key Distribution Center

Native API

Outbound Authentication Message

SOAP Payload

SOAP Payload + Kerberos token

Software AG Training | 15 - 14

Notes:

Kerberos Outbound Authentication – Authentication Mode

- Authentication Mode defined by API Provider
 - Custom Credentials
 - Custom defined values
 - Delegate incoming credentials
 - Option to delegate incoming token or act as normal client
 - Incoming HTTP Basic Authentication
 - Incoming user credentials
 - Incoming Kerberos credentials
 - Incoming Kerberos token to access native API

Outbound Authentication - Transport

Authentication scheme
Kerberos

Authenticate using

- Custom credentials
- Custom credentials
- Delegate incoming credentials
- Incoming HTTP basic auth credentials
- Incoming kerberos credentials

Client password*

Service principal*

Service principal nameform

Username Hostbased

Software AG Training | 15 - 15

Notes:

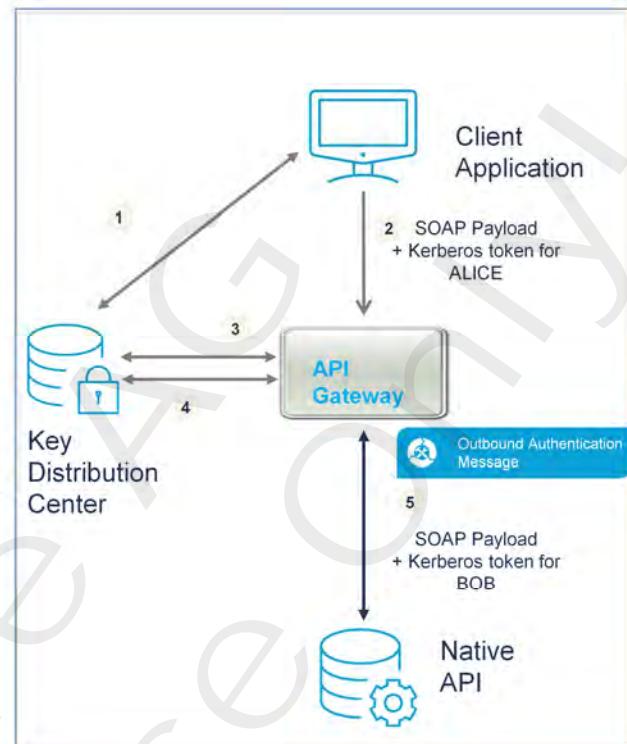
Kerberos Outbound Transport - Policy Configuration UI

- API providers can enforce transport level Kerberos authentication for an API by specifying either the User or Host-based SPN at the outbound

Notes:

Kerberos Authentication Delegation

1. Client requests a Delegated Kerberos token for ALICE
2. Client sends payload and Kerberos token for ALICE to API Gateway
3. API Gateway validates ALICE Kerberos token
4. API Gateway request Kerberos token for BOB using ALICE delegated credentials
5. API Gateway sends request and embedded Kerberos token for BOB to native service



Software AG Training | 15 - 17

Kerberos Delegation is a feature that allows an application to reuse the end-user credentials to access resources hosted on a different server. This means, the kerberos policy is applied in both API Gateway and native server. Client will request delegatable token from KDC, which will be sent to API Gateway. On successful token validation, API Gateway will request kerberos token for clientprincipal configured in outbound policy using the delegated credentials. API Gateway will then send the request to native service with kerberos token in the message.

The screenshot shows the 'Server Configuration' page for the 'WEBMETHODS Integration Server'. The left sidebar has a 'Security' section expanded, with 'Kerberos' selected. The main content area shows 'Kerberos Settings' with the following configuration:

Kerberos Settings	
Realm	
Key Distribution Center Host	
Kerberos Configuration File	C:\krb5.conf
Use Subject Credentials Only	false

A message bar at the top right says 'Kerberos settings saved.' with a link to 'Edit Kerberos Settings'.

Software AG Training | 15 - 18

Server configuration has to be done, for API Gateway to contact KDC, the Key Distribution Center, to fetch Kerberos Token. This configuration has to be done, in IS admin page Kerberos, available under Security section. Kerberos configuration file can be used to be configure Kerberos settings

JSON Web Token: JWT

- **JWT** is a JSON based open standard for creating access tokens that assert a numbers of claims
 - A server could generate a token that has the claim **logged in as admin**
 - Server provides that to a client
 - Client could use that token to prove that (s)he is in logged in as **admin**
- Tokens are designed to be
 - Compact, URL safe, useable in web browser single sign-on (SSO)
- Typical claims
 - Pass identity of authenticated users between an identity provider and a service provider
- Tokens can be authenticated and encrypted
- Available on Inbound and Outbound Authentication Transport

Notes:

Open ID

- OpenID is an open and decentralized authentication protocol promoted by OpenID Foundation
- Allows users to be authenticated by co-operating sites (Relying parties) using a third party service
 - Provides a framework for the communication that must take place between provider and OpenID acceptor (Relying party)
- Available on Inbound Authentication Transport
- API Gateway extracts the ID token from the transport authorization header and validates the token with the claims configured in the application that is requesting access for the API

Notes:



Kerberos Authentication in Message

Notes:

Kerberos Token in a Message

- Similar use cases
 - Inbound
 - Outbound
 - Delegation
- Steps for Inbound Authentication – Message
 - Kerberos Policy is applied in API Gateway
 - Client has to request KDC for token
 - Clients send the token provided by KDC via SOAP message
 - API Gateway during policy execution verifies with KDC for token validity
 - On successful validation request will be routed to the configured endpoint

Here we see the use case of, Inbound Authentication via kerberos token in message. This means, the kerberos policy is applied in API Gateway. client has to request Key Distribution Center KDC for token, and send the token provided by KDC, to API Gateway via the soap message. API Gateway during policy execution will verify with KDC for the token validity. On successful validation, request will be routed to the configured routing endpoint.

Inbound Authentication - Message

Binding Assertion

Require Encryption
+ Add require encryption

Require Signature
+ Add require signature

Token Assertions

Require X.509 Certificate
 Require Kerberos Token
Service Principal Name Form
• Username ○ Hostbased

Require WSS Username token

Service Principal Name * :spartans/anudev.sag.vmchinadfs20w.com

Service Principal Password * :*****

Require SAML Token

Custom Token Assertion
+ Add

Require Timestamp

Software AG Training | 15 - 23

Here is a glimpse of the Inbound policy configuration. Kerberos inbound is part of policy action, Inbound Authentication message. Require kerberos Token check box, has to be enabled, for the configuration to be rendered. Service principal nameform , Service principal name and Service principal password have to be configured. Require time stamp checkbox also has to be selected as mandatory for kerberos inbound.

Kerberos Outbound Authentication Message

- Kerberos has to be selected as authentication scheme
 - Using Incoming HTTP Basic Auth credentials:
 - service principal and nameform has to be configured
 - Client principal has to be passed as part of HTTP basic auth credentials by client, which gateway has to use on inbound
 - Using Custom credentials:
 - client principal, password, Service principal and nameform has to be configured

Software AG Training | 15 - 24

Kerberos outbound is part of policy action Outbound Authentication message. Kerberos has to be selected as authentication scheme, for the configuration to be rendered. Using client credentials, Client principal and password Service principal and nameform , have to be configured. Using Incoming HTTP basic auth credentials, Service principal and nameform , have to be configured, Client principal and password have to be passed, as part of HTTP basic auth credentials by client to gateway, which gateway has to use in outbound.

Kerberos Delegation Message

- Kerberos Delegation is a feature that allows an application to reuse the end-user credentials to access resources hosted on a different server
 - Kerberos Policy is applied in both: API Gateway and native server

Software AG Training | 15 - 25

Kerberos outbound is part of policy action Outbound Authentication message. Kerberos has to be selected as authentication scheme, for the configuration to be rendered. Using client credentials, Client principal and password Service principal and nameform , have to be configured. Using Incoming HTTP basic auth credentials, Service principal and nameform , have to be configured, Client principal and password have to be passed, as part of HTTP basic auth credentials by client to gateway, which gateway has to use in outbound.



SAML Authentication/Authorization in Message

Notes:

Introduction to SAML

- **Security Assertion Mark-up Language** is an XML-based, open-standard data format for exchanging authentication and authorization data between parties, in particular, between an identity provider and a service provider
- Participants
 - SAML authentication involves the following participants



Software AG Training | 15 - 27

Notes:

Subject Confirmation Methods

- Subject confirmation method is used by the relying party to validate the SAML token possessed by the client
- The subject Confirmation methods that are supported in API Gateway are as follows
 - Bearer
 - Holder of Key – Symmetric
 - Holder of Key - Asymmetric

Software AG Training | 15 - 28

Notes:

SAML Inbound - Bearer

- Bearer
 - Client sends request to STS with KeyType as Bearer
 - STS generates SAML Token, signs it with STS private Key, sends token to client
 - Client send request + SAML token to API Gateway
 - API Gateway checks the signature using STS public Key
 - Drawback: no identify check of the client

The diagram shows the flow of the SAML Inbound - Bearer process. It starts with a 'Client Application' sending a request to a 'Secure Token Service'. The STS generates a SAML Token and sends it back to the Client Application. The Client Application then sends a 'SOAP Payload w/ SAML token' to an 'API Gateway'. The API Gateway performs 'SAML Inbound authentication' and then sends a 'SOAP Payload' to a 'Native API'.

Software AG Training | 15 - 29

HoK assertion solves the problem of not being able to check whether the client is authorized to use the SAML token they supply or not.

Here, the client requests SAML token from STS and STS embeds client identity in the provided token. The token is encrypted (using Mediators public key) so that it can't be manipulated by the client - and signed using STS private key.

Client uses this token to call Mediator (after signing the request with the proof token). Mediator checks this signature and decrypts the token to check if the calling client is the one for whom the token has been issued. If so, it calls the native service.

This is the „symmetric” scenario - where client uses the proof key to sign and encrypt

In the asymmetric scenario of HoK the client uses its private key to sign and mediator public key to encrypt.

All the rest is almost the same as with symmetric

SAML Inbound – HOK Symmetric

- Symmetric Key
 - STS generates a proof key as client identity. Embeds the proof key in the SAML token, encrypts and signs the token with its privKey. Sends back token and proof key.
 - Client uses proof key to sign the request. Sends request with embedded SAML token and also the proof key to API Gateway
 - API Gateway validates the signature of the SAML token using STS pubKey, fetches the proof key by decrypting using API Gateway privKey. Validates whether signature and proof key matches.

- Asymmetric Key
 - Same as above, but client uses private key to sign, API Gateway's public key to encrypt

Software AG Training | 15 - 30

HoK assertion solves the problem of not being able to check whether the client is authorized to use the SAML token they supply or not.

Here, the client requests SAML token from STS and STS embeds client identity in the provided token. The token is encrypted (using Mediators public key) so that it can't be manipulated by the client - and signed using STS private key.

Client uses this token to call Mediator (after signing the request with the proof token). Mediator checks this signature and decrypts the token to check if the calling client is the one for whom the token has been issued. If so, it calls the native service.

This is the „symmetric” scenario - where client uses the proof key to sign and encrypt

In the asymmetric scenario of HoK the client uses its private key to sign and mediator public key to encrypt.

All the rest is almost the same as with symmetric

SAML Outbound Authentication

- Act as Client
 - Client sends request to API Gateway
 - API Gateway authenticates the request using any security policies configured in the inbound
 - API Gateway sends SAML token request to STS
 - STS creates SAML token. Sends token to API Gateway.
 - API Gateway embeds the generated SAML token to the request. Sends the request to native API
- Act as Delegation
 - Client calls API Gateway providing its SAML token
 - Client asks API Gateway to Act As the client
 - API Gateway asks STS for an appropriate token

The diagram illustrates the SAML Outbound Authentication process. It shows the flow from a Service Consumer to an API Gateway, which then interacts with a Service Token Service and a Native API. The steps are numbered 1 through 5:

1. SOAP Payload (Service Consumer to API Gateway)
2. Any Inbound authentication (API Gateway)
3. SAML Outbound authentication (API Gateway to Service Token Service)
4. SAML token exchange (Service Token Service to API Gateway)
5. SOAP Payload + SAML V2 token (API Gateway to Native API)

Notes:

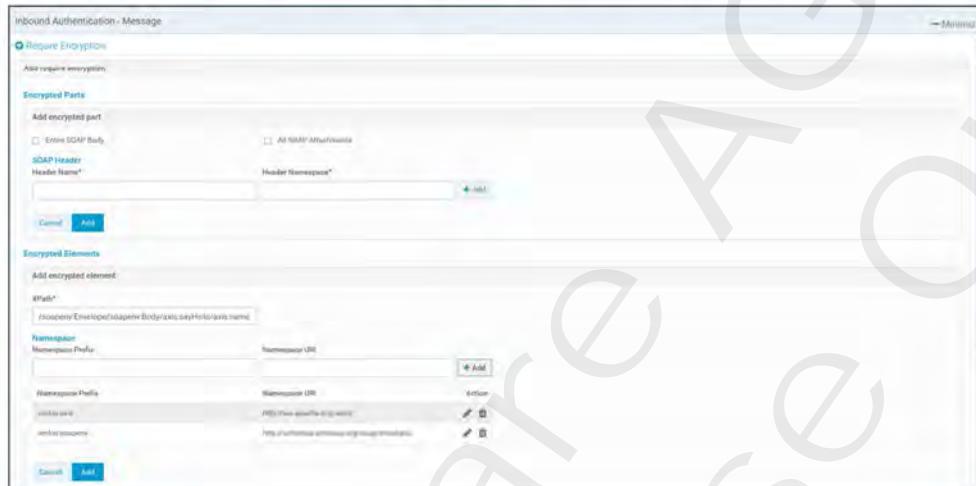


Message Encryption/Signing

Notes:

Inbound Authentication Message - Require Encryption

- WS-SecurityPolicy compliant → can not co-exist with Enable REST Support
- Encrypt/Decrypt data to prevent interception (Confidentiality)
- Requires the incoming message to be encrypted partially or completely



Software AG Training | 15 - 33

Encrypting soap body produces a corrupt soap message because there would not be a soap:Body element in the message.

Provides end-to-end security to protect data from unauthorized access
 Secure and non-secure data can be exchanged in the same document
 Encrypting whole file (not supported)
 Encrypting a single element
 Encrypting the content of an element (not supported)

Inbound Authentication Message - Require Signing

- WS-SecurityPolicy compliant → can not co-exist with Enable REST Support
- Ensure data has not been modified during transport
- Requires the incoming or outgoing message to be signed partially or completely

The screenshot shows the 'Inbound Authentication - Message' configuration screen. The 'Require Signature' tab is active. The interface includes sections for 'Signed Elements', 'XPaths', 'Namespace Prefixes', and 'Signed Parts'. Under 'Signed Elements', there is a table with columns for 'Namespace Prefix' and 'Namespace URI'. Under 'XPaths', there is a table with columns for 'XPath' and 'Action'. Under 'Namespace Prefixes', there is a table with columns for 'Namespace Prefix' and 'Namespace URI'. Under 'Signed Parts', there is a table with columns for 'Signed Part' and 'Add'. Buttons for 'Save' and 'Cancel' are at the bottom.

Software AG Training | 15 - 34

<http://www.w3.org/TR/xmldsig-core/>

Message can contain one or more <sp:SignedElements/> with XPath expressions
Response contains a digital signature element in the SOAP security header to verify

that all elements requesting the XPath expression were signed

Request containing elements which are not signed or no present signature is rejected

Must sign before encrypting elements

Encrypt before sign protection order not supported



OAuth 2 Support

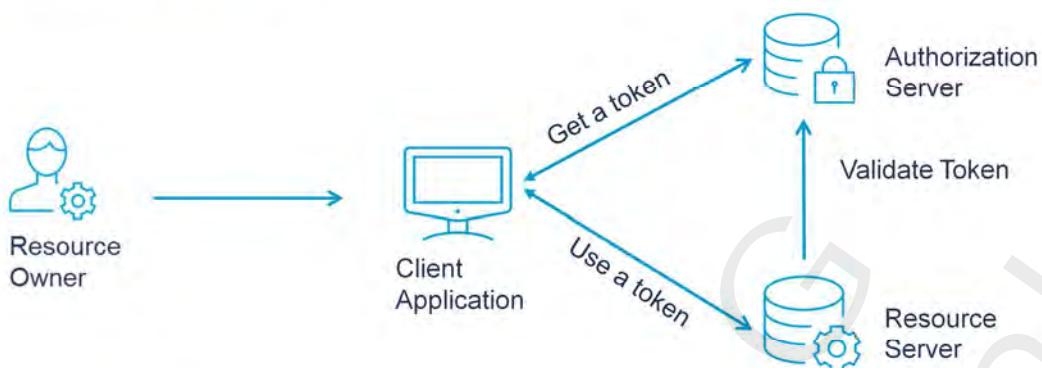
Notes:

What is OAuth?

- OAuth 2 is an authorization framework that enables applications to obtain limited access to user accounts
- It works by delegating user authentication to the service that hosts the user account, and authorizing third-party applications to access the user account.
- Functionality
 - Clients can access server resources on behalf of a resource owner
 - Resource owners can authorize client access to their resources without sharing their credentials
 - Resource owners can limit scope and time of access

Notes:

OAuth – Roles/Actors



- **Resource Owner**
 - End user who can issue grants to a protected resource. Should be a valid user in the authorization server
- **Resource Server (RS)**
 - The server that holds the resource. API Gateway is the resource server which holds the resources/services)
- **Client Application**
 - Consumer application that is registered with API Gateway
- **Authorization Server (AS)**
 - The server that authorizes the client application to access the protected resource

Software AG Training | 15 - 37

Notes:

OAuth Support in API Gateway / API Portal

- Authorization grant is given to a client application by the resource owner, in coordination with the authorization server associated with the resource server
- OAuth 2 lists 4 different types of authorization Grants:
 - Authorization Grant
 - Implicit Grant
 - Client Credentials
 - Resource Owner Password credentials
- Authorization and Access token URL can be
 - HTTP
 - HTTPS
- Tryout OAuth2 protected API in API Portal

Authorization Code

The resource owner (user) accesses the client application.

- 2) The client application tells the user to log in to the client application via an authorization server.
- 3) To log in via the authorization server, the user is redirected to the authorization server by the client application. The client application sends its client ID along to the authorization server, so the authorization server knows which application is trying to access the protected resources.
- 4) The user logs in via the authorization server. After successful login the user is asked if she wants to grant access to her resources to the client application. If the user accepts, the user is redirected back to the client application.
- 5) When redirected back to the client application, the authorization server sends the user to a specific redirect URI, which the client application has registered with the authorization server ahead of time. Along with the redirection, the authorization server sends an authorization code, representing the authorization.
- 6) When the redirect URI in the client application is accessed, the client application connects directly to the authorization server. The client application sends the authorization code along with its own client ID and client secret.
- 7) If the authorization server can accept these values, the authorization server sends back an access token.

Implicit

The implicit grant is a simplified authorization code flow optimized for clients implemented in a browser using a scripting language such as JavaScript. In the implicit flow, instead of issuing the client an authorization code, the client is issued an access token directly after resource owner authorization.

When issuing an access token during the implicit grant flow, the authorization server does not authenticate the client. The access token may be exposed to the resource owner or other applications with access to the resource owner's browser/mobile app.

Implicit grants improve the responsiveness and efficiency of some clients (such as a client implemented as an in-browser application), since it reduces the number of round trips required to obtain an access token.

Client Credentials

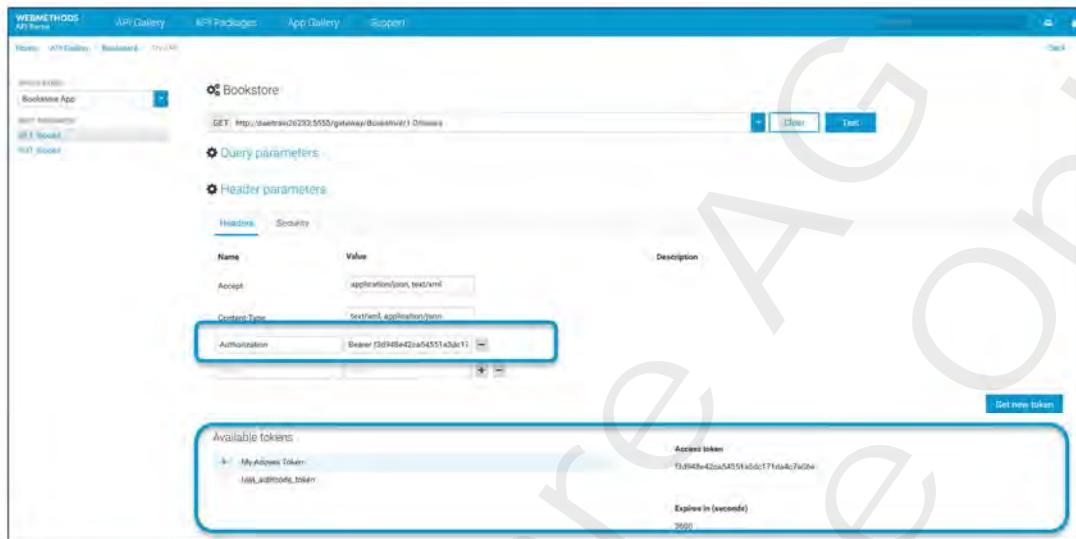
Client credential authorization is for the situations where the client application needs to access resources or call functions in the resource server, which are not related to a specific resource owner (e.g. user). For instance, obtaining a list of venues from Foursquare. This does not necessarily have anything to do with a specific Foursquare user.

Resource Owner Password Credentials

The resource owner password credentials authorization grant method works by giving the client application access to the resource owner's credentials. Using the resource owner password credentials requires a lot of trust in the client application. You do not want to type your credentials into an application you suspect might abuse it. API Gateway doesn't support this grant type.

OAuth 2 Implementation

- Support for registering callback URL in applications
- Fetch and use OAuth2 tokens when testing APIs



Software AG Training | 15 - 39

Notes:

Generation of OAuth2 Token

- OAuth2 Credentials are created as part of an Application requested for the API
- Within the TryOut section of the API on API Portal, the application is listed with the information OAuth2 tokens
 - API Portal TryOut provides support to get a new OAuth2 Token
 - Token name is requested
 - All other properties needed for the request are pregenerated based on the application
- An Approval request from API Gateway is generated to grant access
- In case of approval the access token is generated and provided as Bearer token for the Authorization header of the request

Notes:

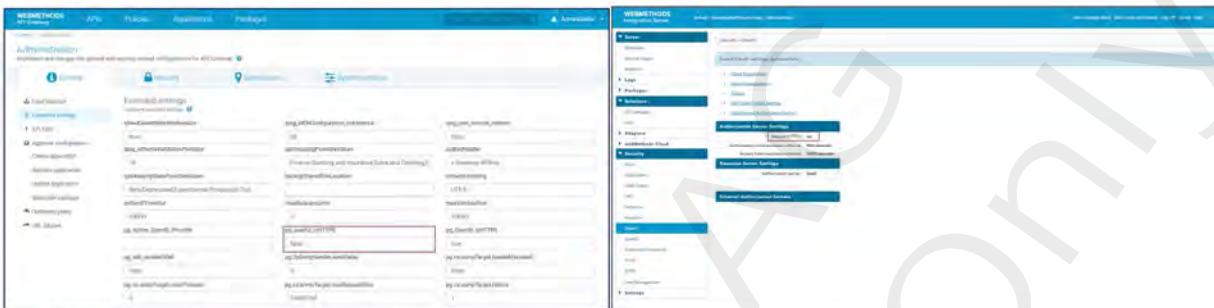
The screenshot illustrates the workflow for generating an OAuth2 access token. It consists of three main panels:

- Top Panel:** Shows the "Processing of OAuth2 Access Token Request" title. Below it, the "Application details" section displays the "Custom Bookstore App" configuration, including the API endpoint `GET /http://devman26233:5555/gateway/Bookstore/1.0/books` and various parameters like "Query parameters" and "Header parameters".
- Middle Panel:** A modal window titled "Get OAuth token" provides detailed information about the token request, including the "Token name" (My address tokens), "Grant type" (Authorization code), "Authentication URL" (`http://devman26233/oauth2/auth.zug.5552/metric/jwt/apikey/authZ/getAddressTokens`), "Access token URL" (`http://devman26233/oauth2/auth.zug.5552/metric/jwt/apikey/authZ/getAddressTokens`), "Client ID" (WebMethods-17b4c34-9493-0ab350f59a18), "Client secret" (a10f2d208-550e-433a-a1a1-1a6b9504e192), and "Scope(s)".
- Bottom Panel:** The "WebMethods API Gateway Resource access approval" screen, which lists the application "Custom Bookstore App-110131a2-9da5-4b57-a280-530e19472ad7 (1.0)" requesting access to the "All" and "Bookstore/1.0 - API Scope" scopes. It includes "Deny" and "Approve" buttons.

Notes:

Configuration in API Gateway and Integration Server

- **HTTP**
 - Set `pg_oauth2_isHTTPs` to `false`
 - Disable **Require HTTPS**



- **HTTPS**
 - Set `pg_oauth2_isHTTPs` to true
 - Enable **Require HTTPS**
 - Create and Enable HTTPS port
 - Make Auth Server SSL enabled and listen to HTTPS port

Software AG Training | 15 - 42

Notes:



Exercise 21

- Try out API secured with OAuth2 token

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



16

API Gateway Administration



Notes:

Objectives

At the end of this chapter you ...

- Know about different Deployment Strategies
- Know about Staging Support in API Gateway
- Know about different Configuration Settings in API Gateway
- Can set up Runtime Aliases in API Gateway
- Know about how to migrate APIs from CentraSite to API Gateway

Notes:

Chapter Contents

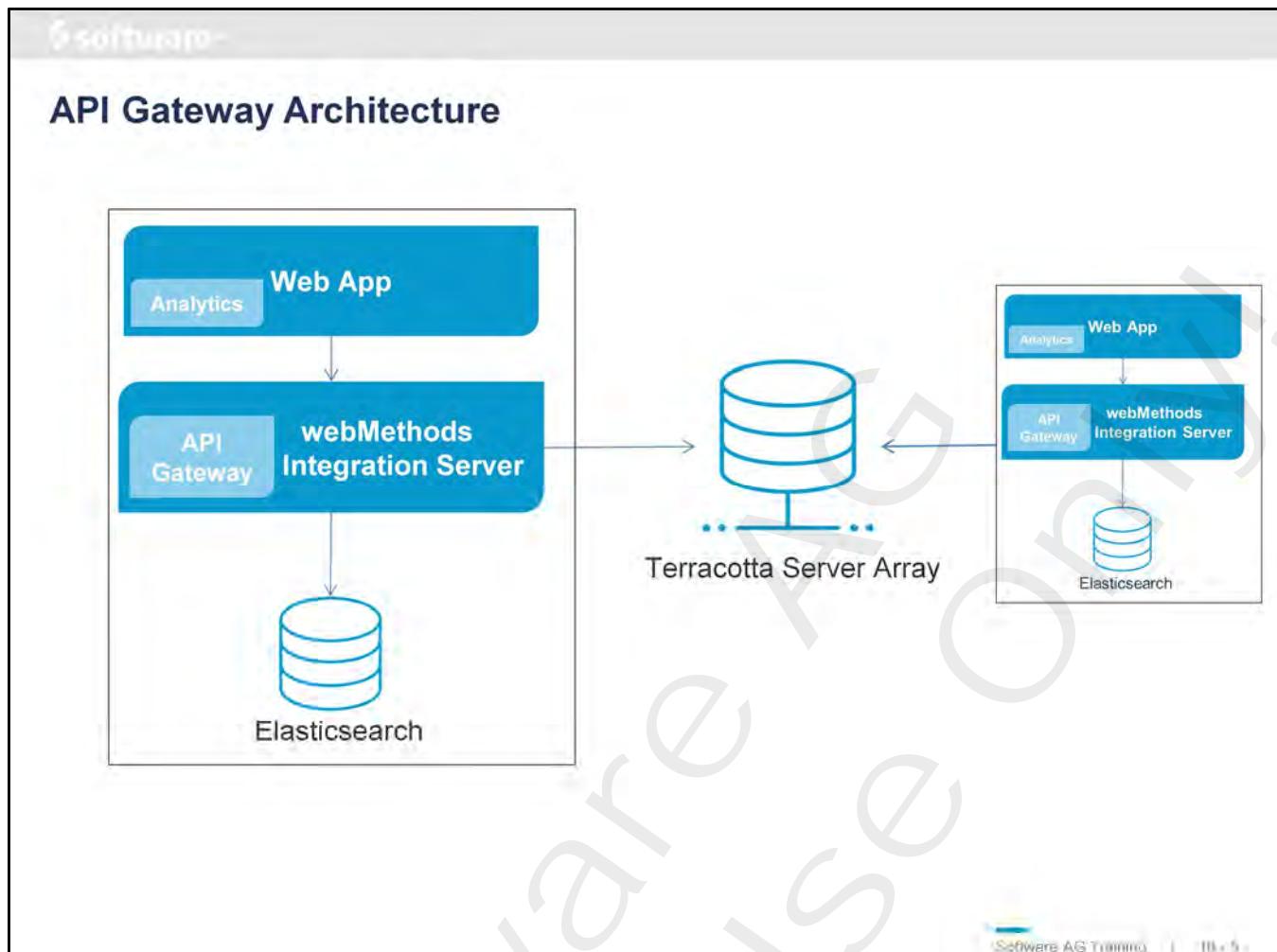
- Architecture
- Deployment
- Staging
- Settings in API Gateway
- URL Aliases
- Migration

Notes:

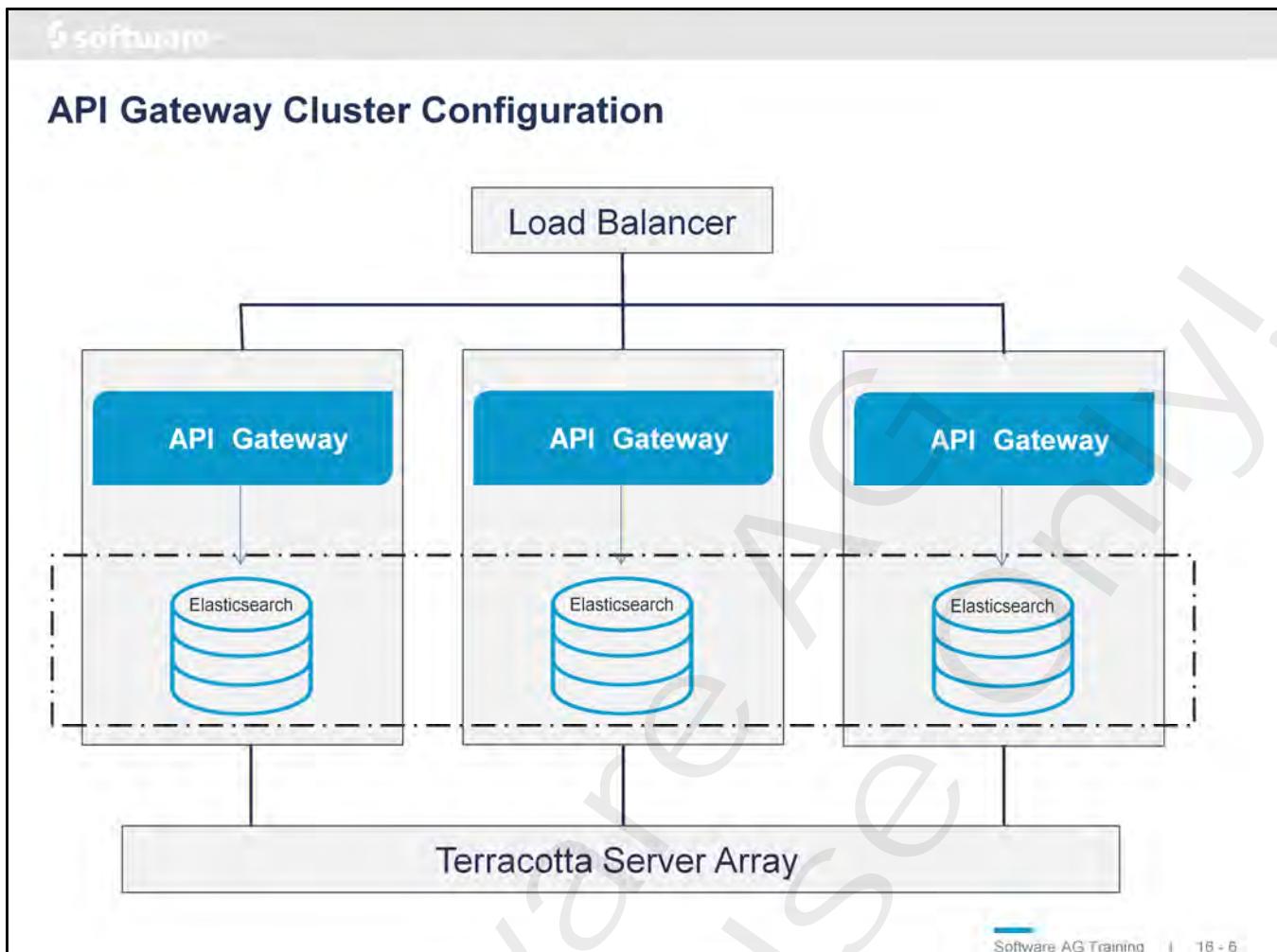


Architecture

Notes:



Notes:



Software AG Training | 18 - 5

Each API Gateway cluster node holds all the API gateway components including UI, the API Gateway package running in webMethods Integration Server and an Event Data Store instance for storing assets-

A load balancer distributes the incoming requests to the cluster nodes. The synchronization of the nodes is performed through a Terracotta Server array and Event Data Store clustering that also has to be defined across the Event Data Store instances. Since each node of an API Gateway cluster offers the same functionality, nodes can be added or removed from an existing cluster. The synchronization of any new node happens automatically. The synchronization covers configuration items and runtime assets like API Policies and applications. The synchronized runtime assets become active automatically.

An event data store instance is a non clustered Elasticsearch node. They should be clustered by modifying the SAG-ROOT/EventDataStore/config/elasticsearch.yml file on each instance using the standard Elasticsearch clustering properties. The cluster name has to be specified and the cluster nodes have to be configured

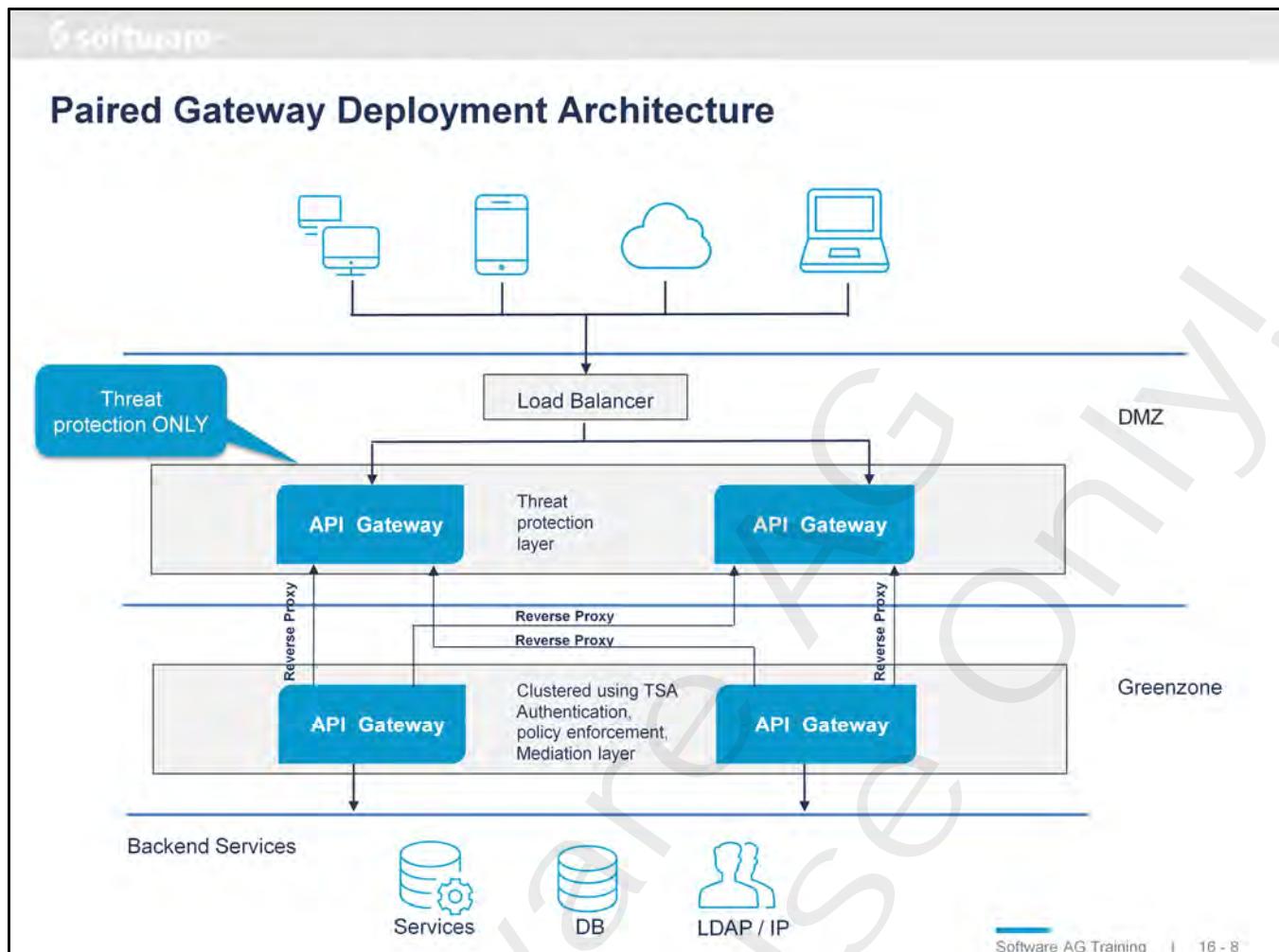
Limitations:

Oauth tokens: are not synchronized across cluster nodes. Currently no workaround
 Parts configuration: are not synchronized. Currently nor workaround.



Deployment

Notes:



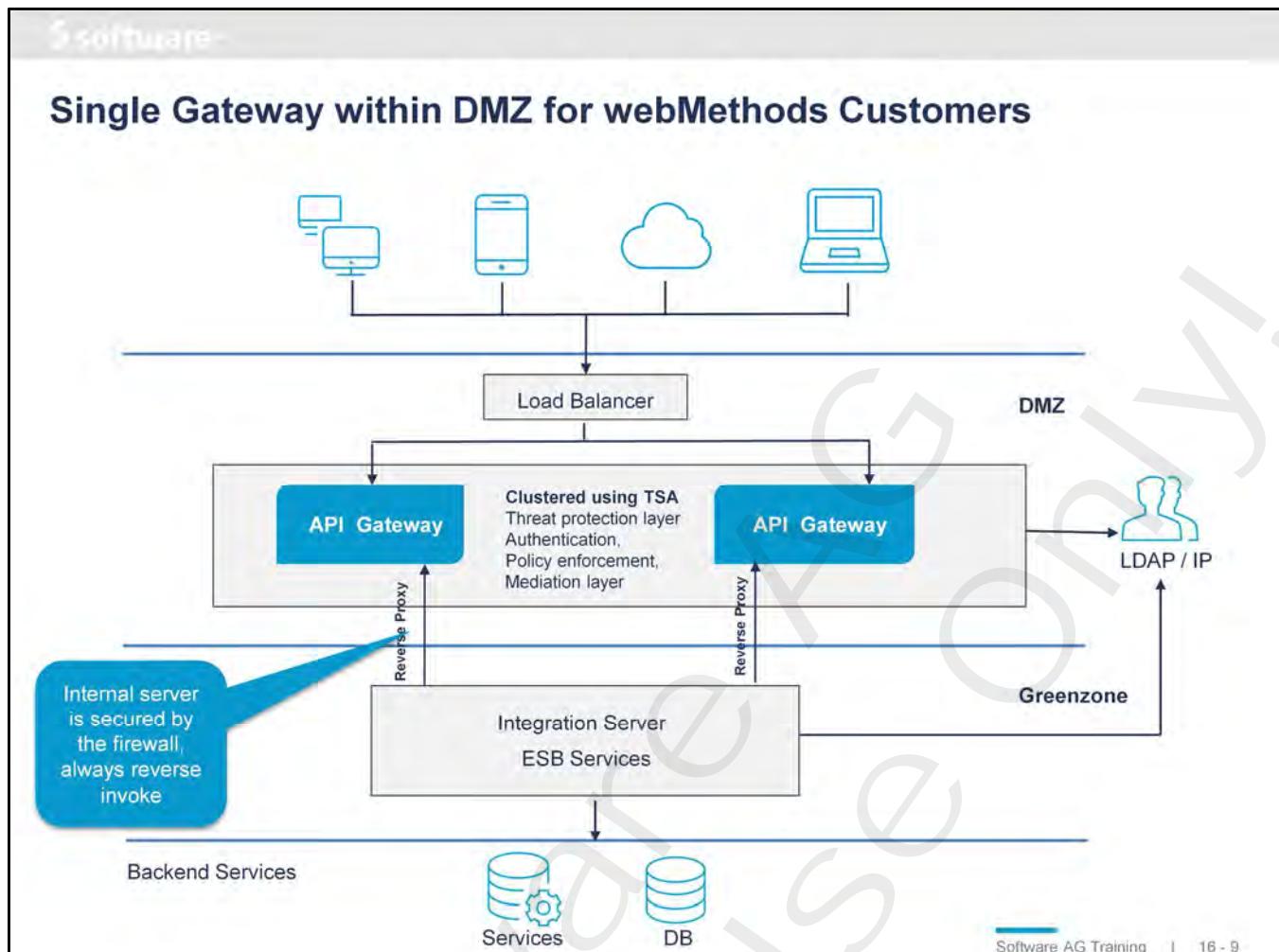
Threat Protection provided in DMZ layer and authentication and policy enforcement in the green zone

Architecture:

1. One API Gateway for threat protection. This layer can have multiple instances using a load balancer
2. One API Gateway for authentication, policy enforcement and mediation. This gateway infrastructure is completely protected using firewall and the communication for DMZ gateway happens through the reverse proxy approach. Various instances of API Gateway can be clustered using Terracotta Server Array

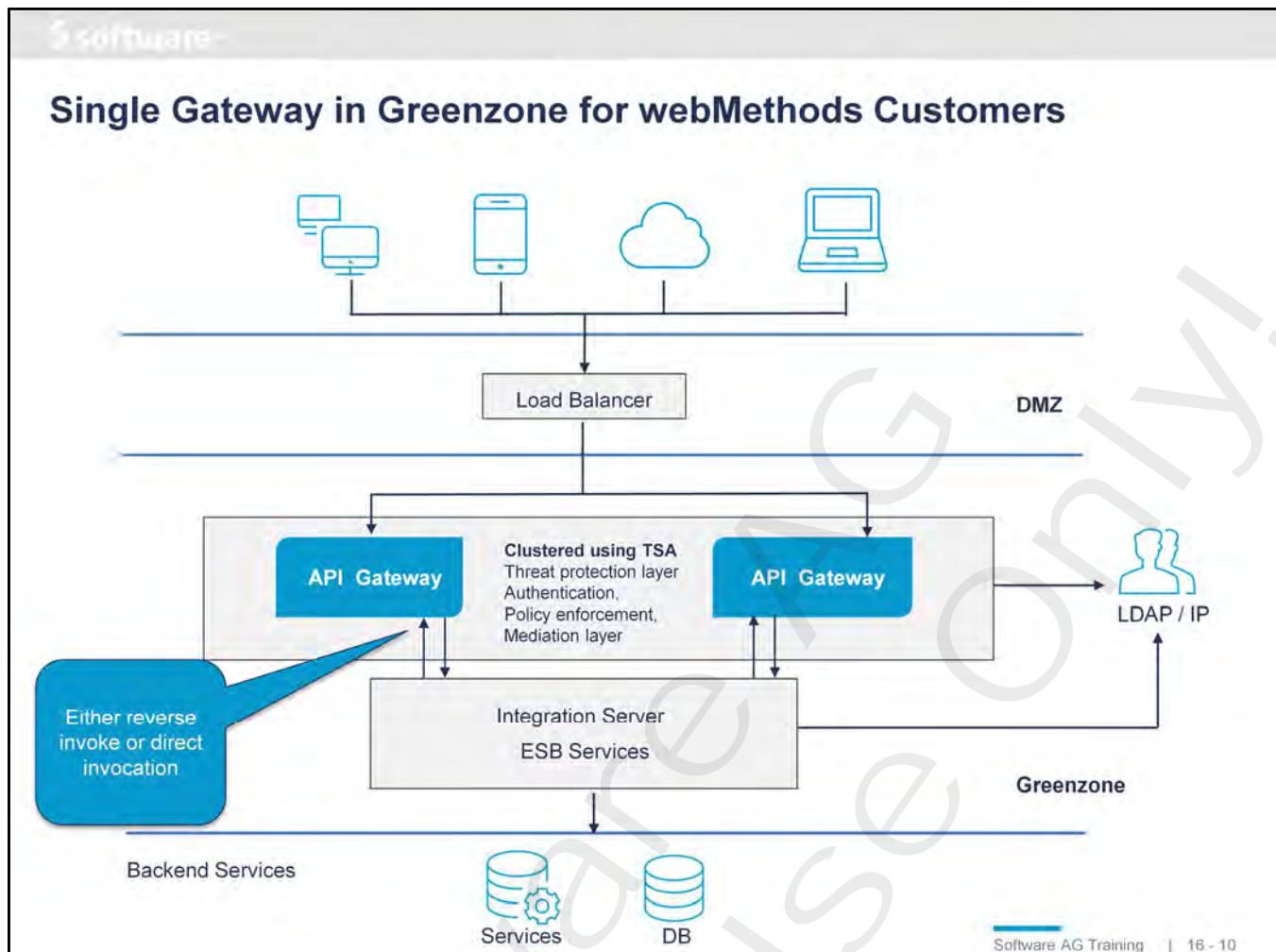
Note:

When changing the rules in one of the API Gateway instances in the DMZ for threat protection you need to restart the other instances to synchronize the rule enforcement



Single API Gateway and the Load balancer in DMZ and IS ESB services in the green zone.

1. Single API Gateway is used for enforcing all policies and rules. There can be multiple instances of API Gateway connected through a load balancer and clustered using Terracotta Server.
2. IS ESB is protected using firewall. API Gateway cannot invoke directly, the endpoint in the routing policy that is applied should be configured as `apigateway://<registrationPort-aliasname>/r<relativepath of the service>`



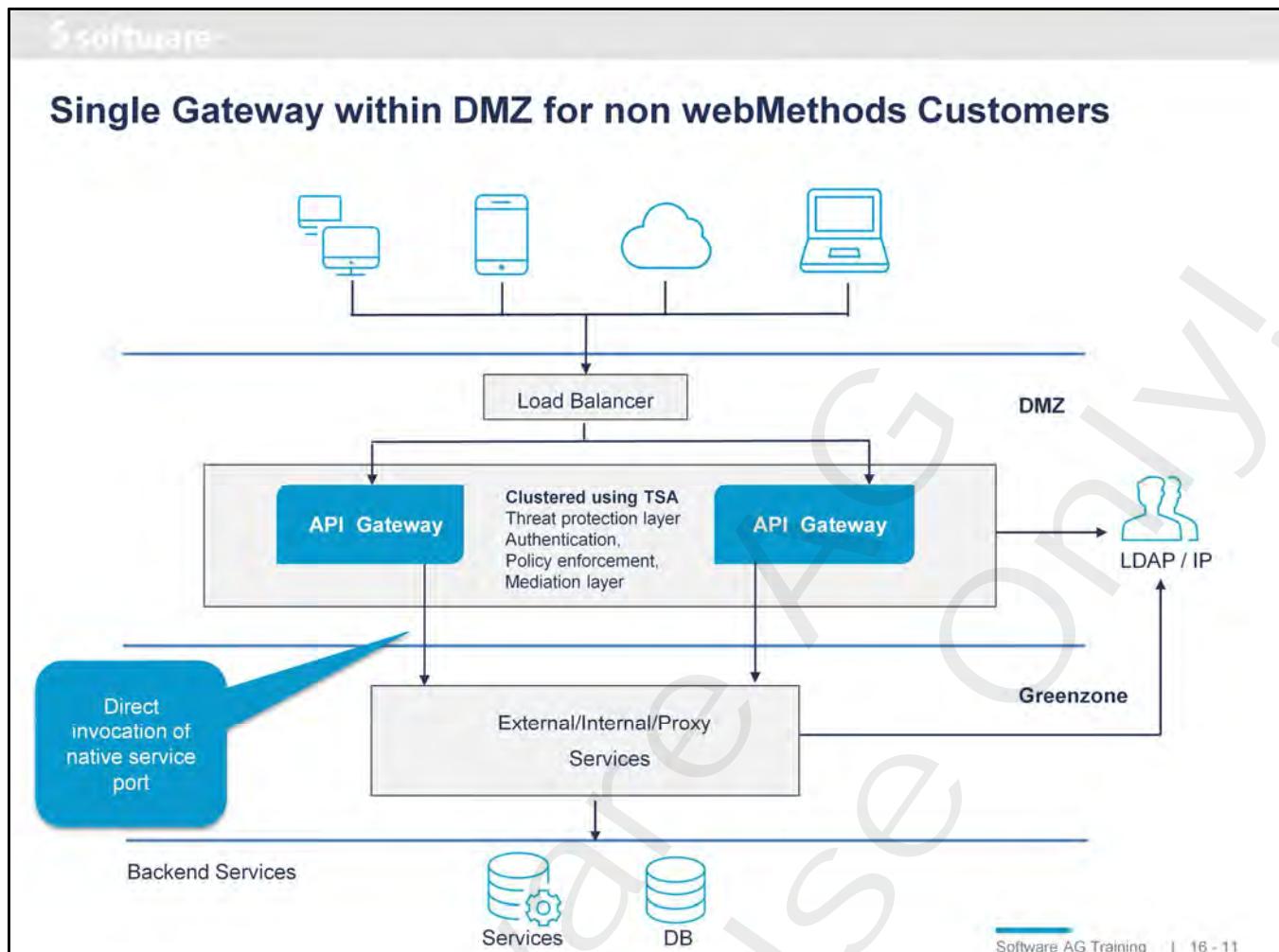
A single API Gateway and native ESB services within the GreenZone and the load balancer in the DMZ

- A single API Gateway is used for enforcing authentication and mediation. Threat protection is not required in this deployment structure, but if required the threat protection rules can be configured for enforcement.
- Multiple instances of API Gateways can be connected through the load balancer and clustered using Terracotta Server.
- There is a direct invocation of ESB services

Note:

API Gateway and ESB services are in the same network.

=> You can either have a direct invocation of ESB services or you can use the reverse invoke approach as required



A single API Gateway and native services within the Green Zone and the load balancer in the DMZ

- A single API Gateway is used to enforce all policies or rules. You can have multiple instances of API Gateway connected through a load balancer and clustered using Terracotta Server Array.
- There is a direct invocation of native services.
- The native service port has to be opened to the mediator network



Staging

Notes:

Staging and Promotion

- Common phases of Service Development Lifecycle (SDLC)
 - Development
 - QA for testing
 - Production for Consumers to consume
- Promotion is the process of moving assets from one stage to another



Software AG Training | 16 - 13

Notes:

Promotion Process

- API Gateway assets
 - APIs, related policies, global policies, applications, packages and plans

- Step 1
 - Build all assets into a repository as a binary object
 - Use webMethods Asset Build Environment (ant based tool)
 - Assets are pulled from the Source API Gateway
 - Assets are converted into binary data (zip file containing APIs, Policy definitions, dependent assets)

- Step 2
 - Deploy the binary build to the target
 - Use webMethods Deployer
 - Define: select assets from to list all available assets in the binary file
 - Map: map the target API Gateways to the set of assets selected
 - Deploy: deploy assets to targets, rollback previous deployment, create snapshots



Software AG Training | 16 - 14

API Gateway assets are mainly APIs and the related policies , Global policies , applications, packages and plans.
Let us now see how we could promote the above assets from one API Gateway to other API Gateway.

Our promotion process is a two step one.

In the first step we build all our assets into a repository as a binary form. This build step is done with webmethods Asset build environment called ABE which is a simple ant based tool .

ABE will contain all the scripts that are needed to build the binary repository for API Gateway.
This is how ABE works for API Gateway.

On execution of the script, ABE pull the assets from the source API Gateway whose connection properties it gets as input. It then converts this asset into a binary data which is nothing but a zip containing the APIs,Policies definitions and their dependencies.

The next step is to deploy the binary build to the Target API Gateways.

The deployment for API Gateway assets is done using webmethods Deployer a tool from SoftwareAG suite that helps in the promotion and staging process of webmethods products.

Deployment using WmDeployer is a three step process.

The first step is to define the repository containing the API Gateway asset binary that we have created using ABE.

Deployer would parse the binary to list all the assets which enables the user to select the assets to be moved to the other stages. The next step would be to map the target API Gateways to the set of assets selected. Deployer also provides the ability for variable substitutions for Specific API Gateway targets.

The final step would be to deploy the assets to the targets. Deployer also provides the user to roll back the last deployment, create snapshots called Checkpoints of the target servers, and delete of assets from the target servers.

With this three step process of Define, Map and Deploy we have moved the assets from the source API Gateway to the target API Gateway.

Continuous Integration and Delivery

- Organizations want to build APIs for easy consumption and monetization
- Enabler for fast to-market initiative
 - Continuous integration
 - Continuous delivery

```
graph LR; A[Automated Provisioning] --> B[Automatic deployments for APIs, policies and other Gateway assets]; C[Continuous delivery] --> D[Integration with Source code management]
```

The diagram illustrates the components of CI/CD. It shows two main paths: one from 'Automated Provisioning' leading to 'Automatic deployments for APIs, policies and other Gateway assets', and another from 'Continuous delivery' leading to 'Integration with Source code management'.

Software AG Training | 16 - 15

In an effort towards modernization of the API and the organization moving towards Agile from traditional waterfall, DevOps is becoming essential in making sure the development and operations team work towards a common goal.

Achieving continuous integration and delivery is key.

As each organization builds APIs for easy consumption and monetization, continuous integration and delivery is integral part of the solution to enable fast to market.

To achieve this for an API perspective two important part are needed.

1. Automated provisioning i.e. Automatic promotions for APIs,policies and other Gateway assets from one stage to other.
2. Integration of the API assets with Source Code management systems like VCS.



Settings in API Gateway

Notes:

General Configuration

- Communication with a Load balancer
- Extended Settings
 - Advanced parameters affecting the operation of API Gateway
 - API Key expiration period, API Maturity state values, API Grouping values, location of the data backup, SMTP trap configurations, configuration of transport protocol for JWT, oauth2, OpenID, ...
- API fault
 - Global service fault settings for errors being returned from API Gateway to the application
- Approval workflow
 - Configuration for creating / updating an application or subscribing to a package
- Configuration of outbound proxy server alias for routing of outbound requests
- URL aliases

Software AG Training | 16 - 17

Notes:

Extended Settings

WEBMETHODS API Gateway APIs Policies Applications Packages Administrator

Home Administration Implement and manage the general and security related configurations for API Gateway.

General Security Destinations System settings

Extended settings

Configure extended settings

allowExceedMaxWindowSize	true	api_grouping_possible_values	Finance, Banking and Insurance, Sales and Ordering, Search, Transport, Beta, Deprecated, Experimental, Production, Test
api_schemaValidationPoolSize	10	api_maturity_state_possible_values	Beta, Deprecated, Experimental, Production, Test
apiKeyHeader	x-Gateway-APIKey	esSerialTimeOut	60000
defaultEncoding	UTF-8	maxWindowSize	10000
maxBackupLimit	-1	pg_oauth2_isHTTPS	false
pg_isJWT_isHTTPS	true	pg_ipSnmpSender.sendDelay	0
pg_isnmpEnableDOM	false	pg_isnmpTarget.maxRequestSize	10485760
pg_isnmpTarget.connTimeout	0	pg_lb_failoverOnDowntimeErrorOnly	false
pg_isnmpTarget.sendTimeOut	500	pg_snmp.communityTarget.maxRequestSize	1
pg_snmp.communityTarget.maxRequestSize	65535	pg_snmp.communityTarget.retries	300

Notes:

Software AG Training | 16 - 18

System Settings

- System level configuration which can be communicated across nodes in the cluster
 - Communication with a Load balancer
 - Dashboard setting
 - Configure a port on which the dashboard runs
 - System setting
 - Configure API Gateway timeout interval
 - Logging setting
 - Modify the logging configuration
 - SAML SSO
 - Configure the SAML settings for single sign on

Administration
Implement and manage the general and security related configurations for API Gateway.

- General**
- Security**
- Destinations**
- System settings**

Configuration

Dashboard setting
Provide the port on which Kiana needs to be started

Port*
9405

System setting
Change the API Gateway system configuration

API Gateway timeout (minutes)
90

Logging setting
Change the API Gateway logging configuration

Log level
OFF

Store log in server

Cancel **Save**

Software AG Training | 16 - 19

Notes:



URL Aliases

Notes:

Alias Definition in API Gateway

- Holds environment-specific property values
 - Can be used in policy routing configuration to provide dynamic behaviour
 - Routing endpoints
 - Routing rules
 - Endpoint connection properties
 - Outbound authentication tokens
 - Outbound HTTP headers
 - at runtime the alias placeholder is substituted with the alias value
- One global definition of the alias value
 - Which is referred to in multiple policies
 - Changing the value of the alias will affect all policies in which it is referred
- Use Case: Staging
 - Moving the API from Test to Production environment just needs an update to the alias definition on API Gateway

Software AG Training | 16 - 21

Notes:

Alias Use Case

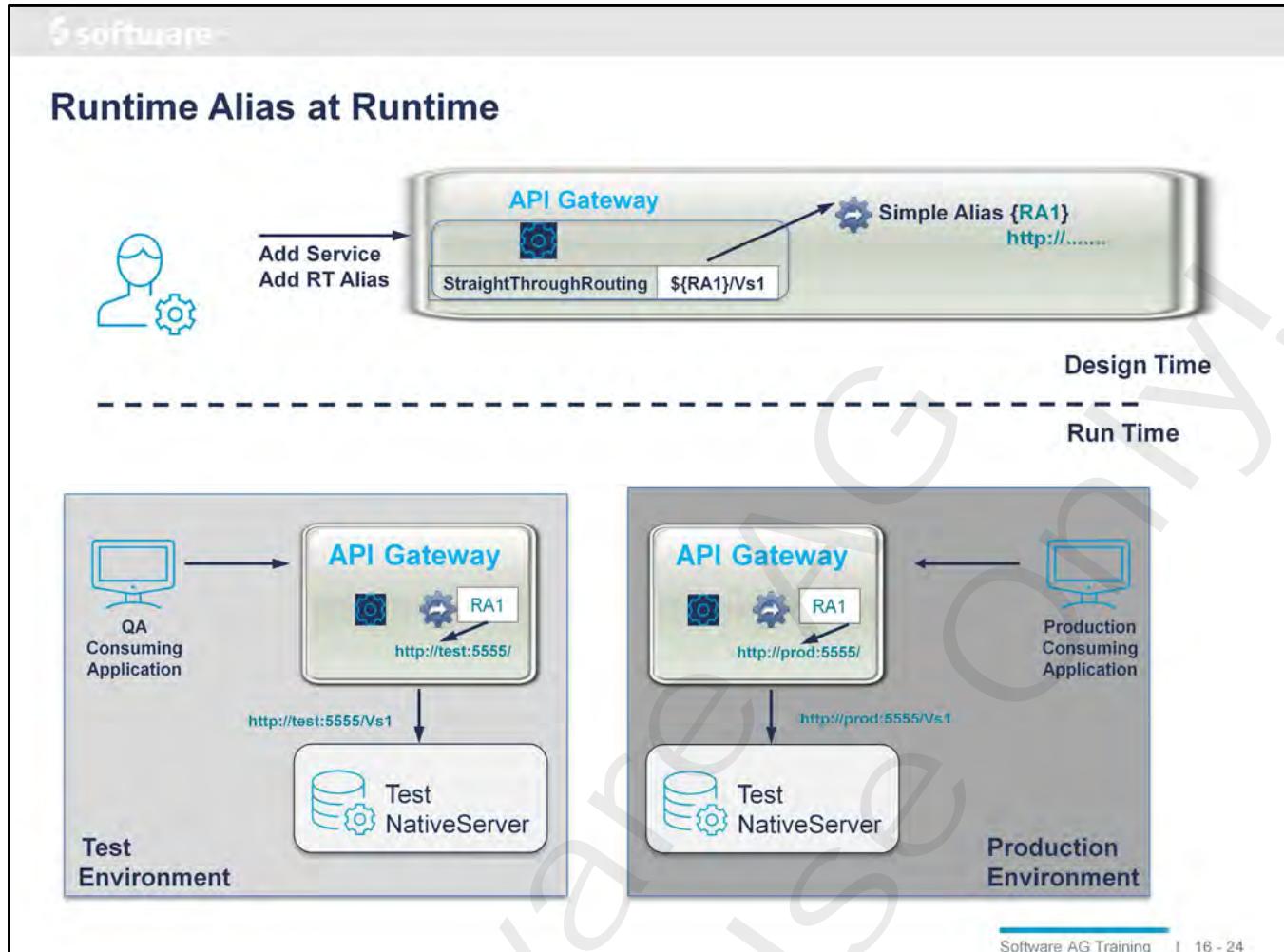
- Dynamic Service configurations in API policies
- Configuration differences in
 - Routing endpoint/destinations for backend/native service
 - Endpoint connection properties like SSL Client Certificate
 - Different outbound authentication tokens
 - HTTP Transport / SOAP Message
- An Alias as an object in API Gateway that holds the above information
 - Separate Runtime Aliases can be defined
 - Policies reference the alias name instead of a hardcoded value
 - at runtime API Gateway dynamically replaces the alias with the alias value as defined in API Gateway

Notes:

Staging Use Case

- Service configurations differ across different staging environments (Development, Test, Production, ...)
- Configuration differences in
 - Routing destinations for backend/native service
 - Different authentication credentials
 - Different endpoint connection properties like SSL Client Certificate
- Promote a service from one stage to the next in API Gateway
 - Manage the routing endpoint for every API in every stage
 - Disadvantage: APIs evolve independent of each other
 - 1 API for all stages and manage stage dependent routing by Context Based Routing Rules
 - Disadvantage: Complex Rule Management
 - Routing Runtime Alias Values reference the actual stage, not the routing in the API
 - at runtime dynamically use the stage specific alias values

Notes:



Notes:

Runtime Alias Types

- Categories for Runtime Aliases
 - Can be routing related URLs -> **Simple Alias**
 - Can be endpoint specific connection properties -> **Endpoint Alias**
 - Endpoint value with additional properties such as connection timeout, whether to pass security information: security headers, keystore alias, key alias, ...
 - Can be transport level security information -> **HTTP Transport Security Alias**
 - **Authentication Scheme:**
 - HTTP Basic authentication
 - OAuth2 authentication
 - NTLM authentication
 - Kerberos authentication
 - **Custom credentials, delegate ...**
 - Can be message level security information -> **SOAP Message Security Alias**
 - **Authentication Scheme**
 - None
 - WSS Username
 - Kerberos
 - SAML
 - **Custom credentials, delegate ...**

Software AG Training | 16 - 25

Notes:



Migration

Notes:

API GATEWAY MIGRATION - STRATEGY

- Migration strategy
 - Based on **CentraSite and API Gateway Integration**
 - Publishing runtime assets(Virtual Services, consumer applications(including API Keys & OAuth2 Clients), runtime policies, runtime aliases) from CentraSite to API Gateway
- Supported Migration flavors
 - Continued usage of CentraSite even after publishing runtime assets to API Gateway.
 - One time publish of runtime assets from CentraSite to API Gateway and then use API Gateway as the source of truth for Runtime assets.

Notes:

API GATEWAY MIGRATION – MODUS OPERANDI

- Mediator Migration to API Gateway – Optional
 - For each Mediator instance, a corresponding API Gateway instance can be provisioned and the configurations migrated using the provided scripts.
 - For detailed documentation please refer “**Migrating Mediator to API Gateway**” section of the API Gateway Configuration Guide
- API Gateway creation in CentraSite
 - Manage Governance Rules in CentraSite Business UI
- Publish Virtual Service(s) to API Gateway
 - Publish specific Virtual Service(s) from API details or search results page in CentraSite Business UI
 - Bulk publish of Virtual Services using CLI
- Verification of Virtual Service(s) in CentraSite & API Gateway - Optional
 - API details in API Gateway
 - API Gateway information profile in CentraSite

Software AG Training | 16 - 28

Notes:

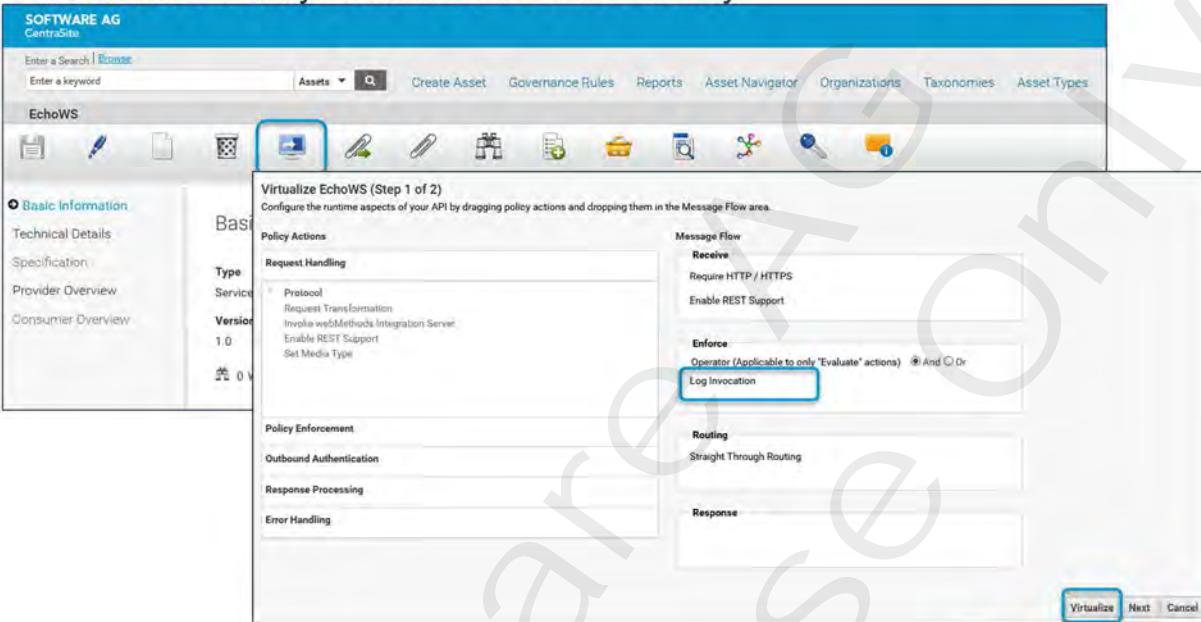
Communication Setup for API-Gateway talking to CentraSite
- CentraSite Endpoint
- User in CentraSite

Communication Setup for CentraSite talking to API Gateway
- API Gateway Endpoint / Integration Server
- API Gateway WebApp URL
- User in API Gateway
- Staging environment

Notes:

Assign Runtime Policies to APIs in CentraSite

- After successful creating and publishing an **API Gateway** target, API Provider can virtualize the REST / SOAP APIs and define Policy Actions – similar set of Policy Actions as in API Gateway



Software AG Training | 16 - 30

Notes:

The screenshot shows the Software AG webMethods Platform interface. At the top, there's a banner with the text "Migration of Assets from CentraSite to API Gateway". Below this, a sidebar on the left lists various service types like Technical Details, Runtime Metrics, Runtime Events, Consumer Overview, Provider Overview, and API Gateway Information. The main area is titled "Basic Information" and shows details for a "Virtual Service" named "EchoWS". It includes fields for Type (Virtual Service), Last Updated (2018-03-07 01:21 PM), Owner (INTERNAL\Administrator), Version (1.0), Organization (Default Organization), and EchoWS(1.0). Below these are metrics: 0 Watchers, 0 Consumers, 0 Consumed Assets, and 0 Pending Approvals. To the right, a "Publish" dialog box is open, showing settings to "Publish to gateway(s) based on the following". Under "All", "All Gateways" and "All Sandboxes" are selected. A table lists "Name", "Type", and "Sandbox" for gateways, with "API Gateway" checked. At the bottom of the publish dialog is a large blue "Publish" button. The bottom right corner of the dialog has a circular arrow icon. The footer of the page reads "Software AG Training | 16 - 31".

Notes:

API Configuration in API Gateway

- API is available as active API on API Gateway
- Policy definitions are the ones defined in CentraSite
 - For policy definitions where destination is configured the information will be retained

Software AG Training | 16 - 32

Notes:

API Gateway Information on API in CentraSite

- Direct link to API detail view on API Gateway:
- Summary of Runtime Enforcement as defined on API Gateway

The screenshot shows the Software AG CentraSite application. In the top navigation bar, there are links for 'Assets', 'Governance Rules', 'Reports', 'Asset Navigator', 'Organizations', 'Taxonomies', and 'Asset Types'. Below the navigation bar, there is a toolbar with various icons. On the left side, there is a sidebar with links for 'Basic Information', 'Technical Details', 'Specification', 'Runtime Metrics', 'Runtime Events', 'Consumer Overview', 'Provider Overview', and 'API Gateway Information' (which is selected). The main content area is titled 'API Gateway Information' and contains sections for 'Display Runtime Enforcements from API Gateway' (Transport, Identity & Access Management, Routing Policies), 'View Runtime Enforcement in CentraSite' (with a blue box around it), and 'View Runtime Enforcement in API Gateway'.

Software AG Training | 16 - 33

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



17

Customization in API Portal

Notes:

Objectives

At the end of this chapter you ...

- Know how to change API Properties in API Portal
- Know how to customize the branding of API Portal

Notes:

Chapter Contents

- API Editing
- API Portal UI Customization

Notes:



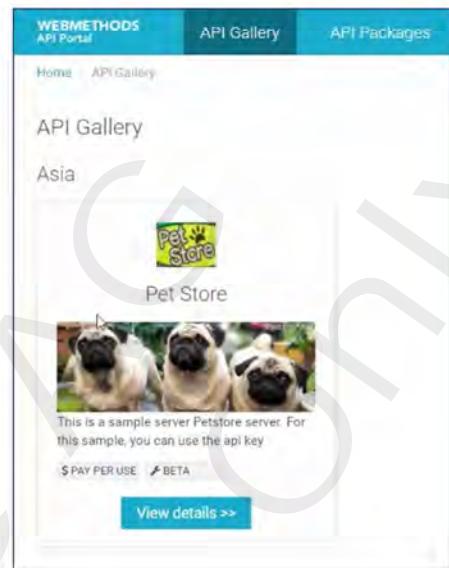
API Editing

Notes:

Software AG Training

API Editing in API Portal

- Purpose is to make your API more
 - Readable
 - Understandable
 - Attractive
- You can change the following Attributes
 - Description
 - Classification
 - Icons
 - Attachments



Software AG Training | 17 / 5

Notes:

The screenshot shows the PetStore API documentation page. At the top, there's a navigation bar with links for Home, Documentation, Examples, and Support. Below the navigation, the title "Edit API Details" is displayed. A sub-section titled "Edit API details from the Portal" is shown with a bulleted list. The main content area displays the PetStore API documentation. It includes a "PetStore" section with an "About" button and a note about the DingTalk API package. Below this is an "API resources" section listing endpoints like "/pet", "/pet/{petId}", and "/pet/findByStatus". Each endpoint has a "Click here to edit" link, a "POST" button, and a "PUT" button. The entire page has a watermark reading "Internal Use Only".

Notes:

Software AG Training | 17 - 7

Editing of the API Details Information

- API Details support HTML content and also support Markdown syntax
- Option to easily enhance the properties by pictures, links, Java Code snippets, ...

The screenshot shows the 'About Pet Store' section of the Pet Store API details. It includes a large image of three pugs, a sample server URL, OAuth and API key access options, and a features section.

About Pet Store

This API is used to get information about pets and update their info. This API is used to get information about pets and update their info. This API is used to get information about pets and update their info. This API is used to get information about pets and update their info.

-a href="/documents/rest/links/e707a63c-435b-4401-92d1-0144941d494d?tenantid=defproject;">
-e href="/documents/rest/links/4e61b3ef-99da-4f4e-9e41-2269f8811ce9?tenantid=defproject;">

Sample projects for different languages can be downloaded using the [Download Client](#).

ACCESS VIA

Features

This package provides tools for the following and more:

Notes:

Advanced API Edit

- Within Advanced Edit you have the option to upload
 - Images
 - File
 - Tags
- Move on to Edit Mode
 - Advanced edit is available



Software AG Training | 17 - 8

Notes:

Advanced API Edit

The screenshot shows the 'Advanced API edit' interface. It has three main sections:

- Change Icon:** A window titled 'Advanced API edit' with a 'REST' icon and a 'Save' button. A message below says 'Maximum file size: 1 MB. Supported file types include JPG, JPEG, GIF, SVG, and PNG.'
- Attach documents:** A window titled 'Advanced API edit' showing a file named 'PetStore JAVA.zip' has been successfully added. It includes a 'Browse' button, a list of files ('File Name: PetStore JAVA.zip, PetStore.json, PetStore.raml'), and an 'Add' button.
- Add tags:** A window titled 'Advanced API edit' showing a 'Business terms' section with 'Pay per use' selected. It includes a 'Save' button and a 'Close' button.

A blue speech bubble on the right says: 'You can use existing tags or create new tags'

Notes:

Revert Changes

- Republishing of the API from API Gateway
 - Will keep all the changes
 - Changes take precedence over what is coming from API Gateway

The screenshot shows the WebMethods API Portal interface. At the top, there is a navigation bar with links for Home, API Gallery, API Messages, and Support. Below the navigation bar, the URL 'Home > Manage API' is visible. The main content area is titled 'Manage APIs' with the sub-instruction 'Import and manage your APIs.' A table lists the API details:

Name	Description	Version
Pet Store	This API is used to get information about pets and update their info. This API is used to get information about pets and update their info. This API is used to get information about pets and update the...	1.0.0

On the right side of the table, there is a 'Revert changes' button with a circular arrow icon.

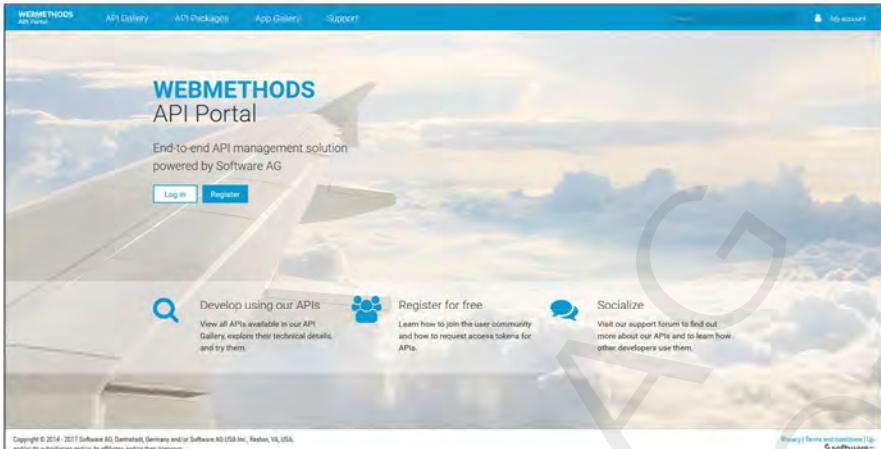
Software AG Training | 17 - 10

Notes:



API Portal UI Customization

Notes:



The screenshot shows the default landing page template for the WEBMETHODS API Portal. The page has a light blue header with the Software AG logo and navigation links for API Portal, API Gallery, API Packages, App Gallery, and Support. Below the header is a large banner with a background image of an airplane wing above clouds. The banner features the WEBMETHODS API Portal logo and a tagline: "End-to-end API management solution powered by Software AG". It includes three main call-to-action sections: "Develop using our APIs", "Register for free", and "Socialize". At the bottom of the banner is a copyright notice: "Copyright © 2014 - 2017 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Herndon, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors."

- The landing page template will be customized by every customer according to their branding
- All images, structure, fonts are changeable
- Anchor templates provided for privacy, terms and support

Software AG Training | 17 - 12

Notes:

The screenshot shows the webMethods API Portal interface. At the top, there's a navigation bar with the 'software' logo, followed by 'WEBMETHODS API Portal', 'API Gallery', 'API Packages', 'App Gallery', 'Support', and 'My account'. Below the navigation bar, the main content area has a heading 'What can be customized?'. To the right of the heading, there's a large, faint watermark reading 'Software AG Internal Use Only!'. The main content lists several ways to customize the portal UI:

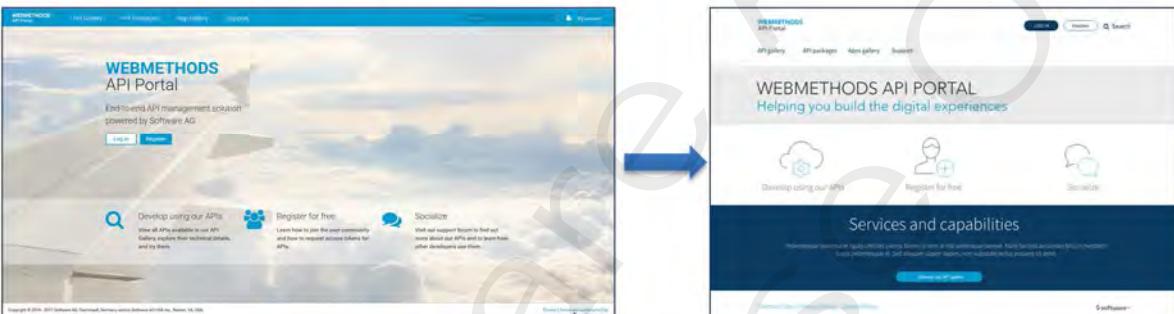
- Portal UI can be completely rebranded to company's look and feel
 - Change
 - Predefined rendering HTML template affecting
 - Layout
 - Font
 - Styles
 - Images
 - Add your own
 - Content blocks
 - Style sheets
 - Java widgets
 - Re-arrange the navigation
 - Include additional links

Software AG Training | 17 - 13

Notes:

Customizing the API Portal

- API Portal UI rendering behaviour is defined using **Views**
- Self Service Configuration capabilities:
 - Create /edit / delete custom view
 - Basic configuration for API Home Page
 - Advanced Configuration (HTML, CSS, Java script knowledge needed)
 - Centralized Resource Management (images, CSS, JS)
 - Backup and restore of views.

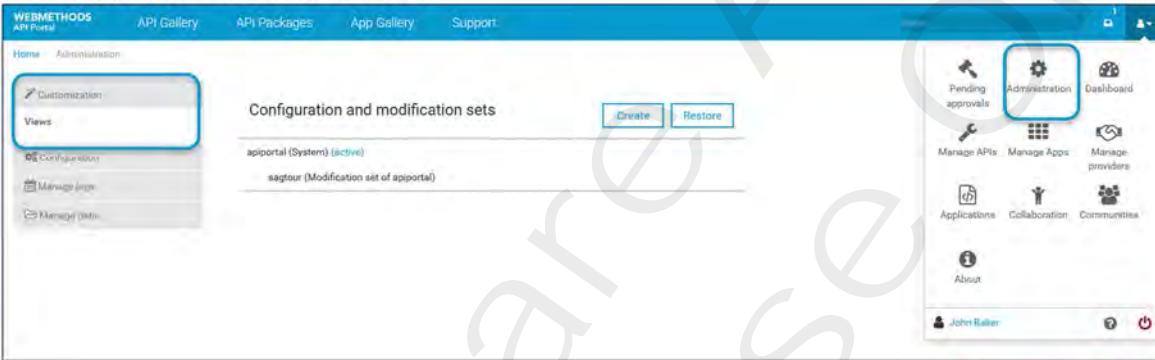


Software AG Training | 17 - 14

Notes:

Administration of API Portal Views

- Customization of API Portal available for
 - API Portal Administrator
- Part of Administration menu
 - Views
 - Create and manage custom view
 - Customize font and color
 - Global definitions for all views

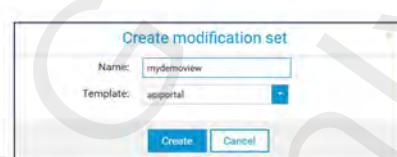


The screenshot shows the 'Administration' section of the webMethods API Portal. On the left, there's a sidebar with 'Customization' and 'Views' selected. The main area displays 'Configuration and modification sets' with a 'Create' button. On the right, the 'Administration' menu is open, with 'Administration' highlighted. Other menu items include 'Pending approvals', 'Dashboard', 'Manage APIs', 'Manage Apps', 'Manage providers', 'Applications', 'Collaboration', 'Communities', and 'About'. The bottom right corner shows 'Software AG Training | 17 - 15'.

Notes:

Creating a New View

- Default Views
 - **apiportal** view, default configuration: **active**
 - sagtours view
- Creating a view leads to a fully functional view
 - Needs a name
 - Must be based on a template
 - As of now only **apiportal** available
- Modify the view
- Activate the view



Software AG Training | 17 - 16

Notes:

The screenshot shows a grid of icons representing different customization options:

- Row 1: About, API details, Footer, Header, Test REST API.
- Row 2: API details, App Gallery, Custom view, Comment stream, Gallery.
- Row 3: Get access token, Home, Detail view for packages, API packages, Privacy settings.
- Row 4: Test SOAP API, Support (highlighted with a blue border), Terms of use, Access tokens, User profile.

Below the grid, there is a list of instructions:

- Select the desired page
- Edit the following sections:
 - HTML
 - JavaScript
 - CSS
 - Image

At the bottom right of the interface, it says "Software AG Training | 17 - 17".

Notes:

Editing a Modification Set - Advanced Configuration

- HTML
 - Modify HTML content
 - Page specific
 - CSS
 - Add/overwrite existing CSS
 - Across pages
 - Image
 - Upload images
 - Can be used in CSS as background image
 - JavaScript
 - To execute any JavaScript function
 - Executed when the page is loaded
 - Across pages

WEBMETHODS API Portal API Gallery API Packages App Gallery Support

Home

```
<div id="api-header-contained"></div>
<div class="home-contained">
<!-- Home Section -->
<!--
Background image has been set by home-bg and home-bg-1, home-bg-2, home-3.css
if you want to set the background to more than 3 or reduce the background image then
set the javascript variable in the javascript section for app
-->
<portal_config_settings>backgroundImage_count = 2;
</portal_config_settings>
<!--
set the css in the css section for app
-->
<home-bg-1>
background-image: url(<@{app.backgroundImage1}>.png);
</home-bg-1>
<home-bg-2>
background-image: url(<@{app.backgroundImage2}>.png);
</home-bg-2>
<!--
-->
<div id="home-app-id" class="home-bg">
</div>

```

HTML

JavaScript

Software AG Training | 17 - 18

Notes:

Customization Menu Options

- Configuration
 - Edit HTML templates, edit/hide Subpages, hide Components

The screenshot shows the WEBMETHODS API Portal interface. A page titled 'Home' is displayed with its source code visible. The source code includes CSS and HTML sections. At the top right of the page, there is a toolbar with several icons. A callout bubble labeled 'View page' points to one of these icons. Another callout bubble labeled 'Advanced configuration' points to a gear icon. A third callout bubble labeled 'Info' points to a small 'i' icon. A fourth callout bubble labeled 'Restore original template.' points to a 'Restore' icon. A fifth callout bubble labeled 'Apply changes' points to a 'Save' icon. A sixth callout bubble labeled 'Back to pages' points to a back arrow icon.

WEBMETHODS
API Portal API Gallery API Packages App Gallery Support

Home

```
1 <div id="api-header-container"></div>
2
18 .home-bg-2{
19   background-image: url(cpn.background-image2.png);
20 }
21 <!-->
22 <div id="home-bg-id" class="home-bg">
23
24
```

Copyright © 2014 - 2017 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

View page

Advanced configuration

Info

Restore original template.

Apply changes

Back to pages

Software AG Training | 17 - 19

Notes:

Customizing Home Page

- Customize background images, which get updated on each visit on the Home page
- HTML & Javascript
 - No changes
- Image
 - Upload your new images and
- CSS
 - Reference your new images

```

<div id="api-header-container"></div>
<div class="home-container">
  <!-- Header Section -->
<!--
Background image has been set by home-bg and home-bg-1, home-bg-2, home-bg-3.
If you want to set the background to more than 3 or reduce the background
set the javascript variable in the javascript section for e.g:
PORTAL_CONFIG_SETTINGS.BACKGROUND_IMAGE_COUNT = 2;
set the css in the css section for e.g:
.home-bg-1{
  background-image: url(cpn.background-image1.png);
.home-bg-2{
  background-image: url(cpn.background-image2.png);
-->
<div id="home-bg-id" class="home-bg">
</div>

```

Software AG Training | 17 - 20

Notes:

Customizing API Gallery

- Re-arrange the assets in the API Gallery page

The screenshot illustrates the customization of the API Gallery page in the webMethods API Portal. The top part shows a grid of eight API assets, each with a REST icon and a brief description. The bottom part shows the same assets rearranged in a different layout, with arrows indicating the movement of assets between the two views.

Name	Description
AirPort API	AirPort API will return Airport details based on the Airport Code
Cable API	AirPort API will return Airport details based on the Airport Code
Github API	AirPort API will return Airport details based on the Airport Code
AirPort API	AirPort API will return Airport details based on the Airport Code
AirPort API	AirPort API will return Airport details based on the Airport Code
AirPort API	AirPort API will return Airport details based on the Airport Code
AirPort API	AirPort API will return Airport details based on the Airport Code
AirPort API	AirPort API will return Airport details based on the Airport Code

Notes:

Backing Up a Modification Set

- Customization changes / modification set can be saved in the file system
 - As zip file

WEBMETHODS API Portal API Gallery API Packages App Gallery Support

Home | Administration

Customization

Views

Configuration

Manage logs

Manage data

Configuration and modification sets

Create Restore

apiportal (System) (active)

mydemoview (Modification set of apiportal) (active)

sagtour (Modification set of apiportal)

Backup

- Customization changes / modification set can be restored in another tenant or in another instance

WEBMETHODS API Portal API Gallery API Packages App Gallery Support

Home | Administration

Customization

Views

Configuration

Manage logs

Manage data

Configuration and modification sets

Create Restore

apiportal (System) (active)

mydemoview (Modification set of apiportal)

sagtour (Modification set of apiportal)

Software AG Training | 17 - 22

Notes:



Exercise 22

- Editing APIs in API Portal

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!



18

API Portal Administration

Notes:



Administration Tool

Notes:

Chapter Contents

- Administration Tool
- Community Support
- Client SDK
- User Onboarding
- Multifactor Authentication
- App Marketplace
- Extension Points

Software

Basic Admin Tooling

- Portal installation creates a shortcut for “API-Portal Cloud Controller”
 - Similar to ARIS Cloud Controller
- Single admin tool for configuration, list, start, stop (kill), reconfigure & monitoring each component
- Example:
 - set services to autostart
 - reconfigure the loadbalancer




Software AG Training | 18 - 4

Notes:



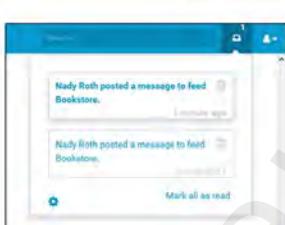
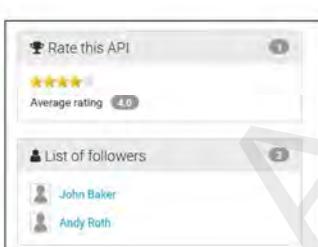
Community Support

Notes:

Software AG

Collaboration

- Sharing of APIs with Google+, Facebook or e-Mail
- Following APIs for updates
- Notifications



SearchCruise REST API

f G+ Share via e-mail

Rate this API

Average rating 4.0

List of followers

John Baker
Andy Roth

Nady Roth posted a message to feed Bookstore.
Nady Roth posted a message to feed Bookstore.

Mark all as read

Software AG Training | 18 - 6

Notes:

API Communities

- Control the exposure of APIs to support B2B scenarios where partners should only see certain APIs

The diagram illustrates the concept of API communities. At the top, three API descriptions are shown: 'SearchAPI' (Search icon), 'Sign-upAPI' (User icon), and 'CruiseMgmtAPI' (Bell icon). Each API has a 'View details >>' button and two status buttons: 'Free' and 'Production'. Below these, two groups of users are represented by icons: 'Cruise Consumers' (three people) and 'Cruise Providers' (three people). Lines connect the 'SearchAPI' and 'Sign-upAPI' to the 'Cruise Consumers' group, indicating they are accessible to them. Lines also connect the 'CruiseMgmtAPI' to the 'Cruise Providers' group, indicating it is accessible to them.

The purpose of API communities is to control the exposure of APIs. This is necessary to support B2B scenarios where certain partners should only see certain APIs. An example for such scenario is shown on this slide that shows the APIs Cruise Bookings, Cruise Searches and Cruise Management. The Members of the Cruise Consumer community have access to the Bookings and Searches API. The Cruise Providers community exposes the Searches and the Management API.

API Communities in webMethods API-Portal

- API community is a group of consumers (members) that can consume a set of associated APIs
- API-Portal support: 1 public community and n private communities
- Public Community
 - APIs associated with the public community are visible to all consumers and to the unregistered users
 - Self registered users become members of the public community
- Private Community
 - APIs associated with a private API community can only be consumed by the community members
 - Private communities can be created and removed by API Administrator
 - API communities have community administrators to manage the community members

To support such B2B use cases the webMethods API-Portal offers API communities which are defined as a set of users that can consume a set of associated APIs.

Existing API-Portal users as well as new users can be invited to a community. New users are running through a simplified onboarding process.

API communities have dedicated members the community administrators. Community administrators have the privilege to manage the community members.

The screenshot shows the webMethods API Portal interface. On the left, a sidebar menu includes options like Pending approvals, Administration, Dashboard, Manage APIs, Manage Apps, Manage providers, Applications, Collaboration, and Communities. A blue arrow points from the 'Communities' option in the sidebar to the 'Communities' tab in the main content area. The main content area displays the 'Create community' screen for a 'Cruise Community'. It shows an 'Overview' section with the name 'Cruise Community' and a description 'Community for exposing APIs related to cruise searching and booking'. Below this is a 'Members' section listing a single member with the name 'customer@company.com', email 'customer@company.com', and an unchecked 'Administrator' checkbox. To the right of the member is a 'Delete' button. At the bottom of the members section is a 'Delete' button. The 'APIs' section lists three APIs: 'SignupAPI', 'SearchCruises', and 'BookingAPI'. For each API, there is an 'Add' button, a 'Version' dropdown set to '1.0', and a 'Delete' button. At the bottom of the APIs section are 'Apply' and 'Cancel' buttons.

Software AG Training | 18 - 9

Notes:

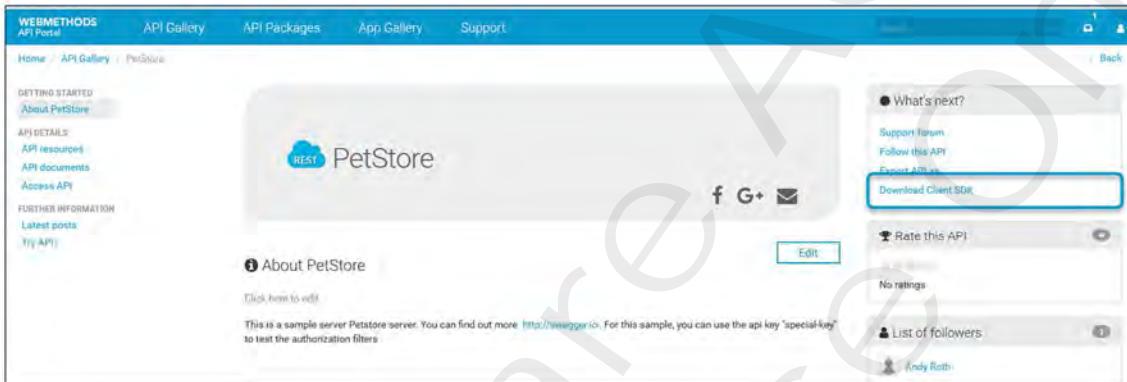


Client SDK

Notes:

Client SDK Generation

- Helps to generate client code quickly and easily
- Facilitates in
 - Testing the API
 - Integration with application
- Provides various language options including widely used languages such as
 - Java, .Net, ...



Software AG Training | 18 - 11

Notes:

Processing Client SDK

- Client SDK will be provided as zip-file

```

    /**
     * Returns a single pet
     *
     * Find pet by ID
     *
     * @throws ApiException
     *         if the API call fails
     */
    @Test
    public void getPetByIdTest() throws ApiException {
        Integer petId = 1;
        Pet response = api.getPetById(petId);
        System.out.println("Response:" + response);
        // TODO: test validation
    }

    /**
     * (Untitled)
     *
     * Get user by user name
     *
     * @throws ApiException
     *         if the API call fails
     */
    @Test
    public void getUserByNameTest() throws ApiException {
        String username = null;
        // User response = api.getUserByName(username);
    }
}

```

Software AG Training | 18 - 12

Notes:



User Onboarding

Notes:

Steps for Simple User Onboarding – Default configuration

- Register User
 - Start Registration Action
 - Provide account details and email and password
- Activate User account
 - Wait for registration confirmation email
 - Use link to activate account in API-Portal
- Login to API-Portal
 - Wait for account activation notification
 - Login to API-Portal with username and password

Software AG Training | 18 - 14

Notes:

Required Levels of Approval in API-Portal

- Required functionality in API Portal for requesting an account when signing up
 - Approval Only Mode
 - Start approval workflow
 - Send user notification (registration requires Approval from API Provider)
 - user has an option to specify a message/reason for signup
 - No Approval Mode, but Email Confirmation Selection
 - Send API Portal User Creation Confirmation
 - User can activate account through email link
 - None
 - Create and activate user

Software AG Training | 18 - 15

Notes:

The screenshot shows the API-Portal Administration interface. On the left, a sidebar menu includes 'Customization' (selected), 'Configuration' (highlighted with a blue box), 'User registration', and 'Configuration settings'. The main content area is titled 'API Portal configurations' and contains several configuration sections:

- LDAP user registration setting:** Set to 'STRAIGHTTHROUGH' with a dropdown menu.
- User registration approver group:** Set to 'Crush Community_admins'.
- Time To Live (in minutes) for the email activation link:** Set to 30.
- Subject for the email to Administrator about failed user request:**
- Message for the email to Administrator about failed user request:**
- API Portal failure [@request]:** Displays a template message with placeholders for recipient name, request details, timestamp, and reason.
- Best Regards,** API Portal Team
- Request could not be processed [@request]:** Displays a template message with placeholders for recipient name, request details, timestamp, and a note to try again later.
- Best Regards,** API Portal Team

The top right corner of the configuration section has a blue box around the gear icon in the navigation bar. The navigation bar also includes 'Pending approvals', 'Dashboard', 'Manage APN', 'Manage Apps', 'Manage providers', 'Applications', 'Collaboration', and 'Communities'.

Software AG Training | 18 - 16

Notes:

User Registration Administration

- Email Confirmation
 - User registers self to API-Portal using the link sent as part of the user's email
 - Validates the email address provided by the consumer
- Approval Only Mode
 - Just approval workflow
 - With Email confirmation
- Automatic Registration
 - None



Define registration process

This page enables the administrator to specify how user registration is handled.

E-mail confirmation required Approval required Automatic registration

Automatically send an e-mail confirmation with the following subject and contents

Subject: Activate your account

Contents:

```
Hello @requestor.name,  
Congratulations! You have successfully registered in API Portal.  
You can activate your account by clicking this link: http://loadbalance.com:8080/#/tenantId/activationId
```

Apply

Software AG Training | 18 - 17

Notes:

User Management

- Configuration of approval for user registration

Home Administration

Customization Configuration

User registration Configuration settings

Manage logs Manage data

Define registration process

This page enables the administrator to specify how user registration is handled.

E-mail confirmation required Approval required Automatic registration

Select workflow approval stages and specify when to automatically send an e-mail notification with the specified subject and contents.

Notifications

Send notification to requester that the request has been received.

Subject: Registration request status update

User group master: API Administrator, API Consumer, API Consumption Approver, Create Community, Create Community Admin, Public Community, Public Community Admin

User group master: API Provider, API User Registration Approver

Add all All remove all

Cancel

WEMETHODS API Portal API Gallery API Packages App Gallery Help

Pending approval requests

User ID: pete@company.com

E-mail address: pete@company.com

Reason for request:

Origin:

Company:

Software AG Training | 18 - 18

Notes:



Multifactor Authentication

Notes:

Software AG

Multifactor Authentication

- Support for multifactor authentication
 - Verification of a user's identity requires 2 or more authentication factors
 - Knowledge factor (password, ...)
 - Possession factor (security token, ...)
 - Inherence factor (biometric verification, ...)
- API Portal uses a combination of username, password and one-time password (OTP) to verify user's identity



The screenshot shows a 'Log in' page with two main sections: 'Log in using your basic account' and 'Log in using your mobile device'. Below these are buttons for 'Create account' and 'Forgot password?'. A watermark reading 'Internal Use Only!' is diagonally across the page.

Software AG Training | 18 - 20

Notes:

Provision of One-Time Password

- User receives OTP via
 - email
 - User can request a new OTP which is sent to the user through email
 - As secret token in an email
 - User can use the secret token and generate am OTP via external token service (Google Authenticator)
- Multifactor Authentication (MFA) turn on
 - On user registration, user receives a secret token through email
 - Use external client to generate an OTP using the secret token
 - You get back a token for this specific user
 - Use OTP to logon onto API Portal

Notes:

Added Token Authentication

- API Portal configured to authenticate against LDAP
- Login with LDAP account
- API Gateway asks for
 - One-time password

The screenshot illustrates the two-step authentication process. On the left, the 'API Portal' login screen shows fields for 'User name' and 'One-time password'. An arrow points from the 'One-time password' field to the right side of the screen. On the right, a separate window titled 'Authenticator' displays the generated one-time password '054770'. A blue speech bubble above the authenticator window is labeled 'External authentication client'.

Software AG Training | 18 - 22

Notes:

Configuration in API-Portal

- Configuration has to be set in User Management Console
- Property: **Use multi-factor authentication**
 - Enable/disable
 - API Portal sends out a secret token to users when on-boarding
- Property: **Exclude users**
 - to list of users
 - API Portal excludes these users from MFA
- Users who were onboarded before enabling MFA
 - Select required user
 - Click **Generate token secret**

Notes:



App Marketplace

Notes:

API Portal – APP Marketplace

- Support for building an ecosystem
- Register and promote Apps using APIs available in the API Portal
- Browse and search Apps
- Rate Apps and collaborate within the API Portal

The screenshot shows a web-based application interface for managing APIs. At the top, there's a navigation bar with links for 'Home', 'API Gallery', 'API Packages', 'App Gallery', and 'Support'. Below the navigation, there's a search bar and a 'Logout' button. The main content area features a card for an app named 'TempleRun'. The card includes a small icon of a temple, the app name, and a 'Delete' button. To the right of the card, there's a sidebar with developer information (email: erik@softwareag.com), supported platforms (Android, iOS), and links for 'What's next?', 'Report issue', 'Follow App', 'List of followers', and 'User Lastname'. Below the card, there's a section for 'Developed using' with a 'SOA' icon and a screenshot of the game Temple Run.

Software AG Training | 18 - 25

Notes:

The screenshot shows the 'Create App' interface in the webMethods API Portal. The app is titled 'TempleRun'. The 'Description' field contains the text: 'In pretty much every treasure hunting adventure movie there's one specific scene in which the plucky hero finally gets his hand on the treasure but then has to navigate a maze of booby traps to get out alive. And it's amazing.' - SlideToPlay.com. The 'Features' field includes the text: 'Best endless running game in the App Store ... You'll love every minute.' Below the description, there is an 'Icon' section with a placeholder image and a 'Browse' button. The 'Version' is listed as '1.6.3'. The 'Company' is 'Imangi Studios, LLC'. There are also sections for 'Tags' and 'Notes'.

Software AG Training | 18 - 26

Notes:



Extension Points

Notes:

API Portal Extension Points

- API-Portal is configurable to work as a standalone component and can be integrated with any third party API-Management systems, which may not necessarily have **API Gateway** or **CentraSite**.
- In a Standalone deployment, API-Portal sends an email to a group of configured users for access token management.
- Deploying API-Portal in a third party environment is done in two steps:
 - Managing third party key management providers
 - Enable API-Portal Extension Point, Adding User to API Consumption Approvers Group, Request Access Token
 - Managing Access Tokens
 - Requesting access token, Mailing email templates, Provisioning access tokens

Software AG Training | 18 - 28

API-Portal is configurable to work as a standalone component and can be integrated into any third party API-Management systems, which may not necessarily have Mediator or CentraSite present.

You can configure API-Portal to work with different key management systems for access token retrieval functions for the APIs deployed in the third party environments.

API-Portal sends email to all the users in the UMC group API Consumption Approvers for access token requests.

Deploying API-Portal in a third party environment is done in the following stages:

Managing third party key management providers

Managing Access Tokens

API Portal Extension Points

- API-Portal is configurable to work as a standalone component and can be integrated with any 3rd party API-Management system, which may not necessarily have **API Gateway** or **CentraSite**.
 - Extension Points provide easy integration of 3rd party solutions with Software AG API-Portal
 - Customers don't need to talk to their 3rd party API providers to integrate with Software AG API-Portal. They can use API-Portal itself to integrate.
- Different groups of Extension Points APIs
 - API Repository APIs – Basic and API Metadata
 - API Provider Registration APIs
 - API Related Events APIs
 - API Packages and Plans APIS – Manage and Associate

Software AG Training | 18 - 29

API-Portal is configurable to work as a standalone component and can be integrated into any third party API-Management systems, which may not necessarily have Mediator or CentraSite present.

You can configure API-Portal to work with different key management systems for access token retrieval functions for the APIs deployed in the third party environments.

API-Portal sends email to all the users in the UMC group API Consumption Approvers for access token requests.

Deploying API-Portal in a third party environment is done in the following stages:

Managing third party key management providers

Managing Access Tokens

Managing 3rd Party Key Management Providers

- API-Portal logs events related to access token (key requests, key renew, key revoke)
- How to provide an access token for a consumer from an external provider ?
 - Access the list of all events status INPROGRESS
 - Publish the access token into API Portal

Function	Method	Resource path
Get the list of all the active events	GET	/abs/apirepository/v1/events
Get the list of all active events for a list of API	POST	/abs/apirepository/v1/events/apis
Change the event state on successful poll	POST	/abs/apirepository/v1/events
Publish the key into API Portal	POST	/abs/apirepository/v1/accesstokens?eventId={eventId}
Renew the key in the API Portal	PUT	/abs/apirepository/v1/accesstokens/{tokenuuid}?eventId={eventId}
Delete the key from API Portal	DELETE	/abs/apirepository/v1/accesstokens/{tokenuuid}?eventId={eventId}

Software AG Training | 18 - 30

Notes:



19

Wrap Up

Notes:

PRIME Value Creation Methodology

Prime is the **solution deployment** methodology to create **maximum value** using the Digital Business Platform

Methodologies
for your
repeatable Project Success

60+ Service Packages

700+ WORK
PACKAGES

500+ ACCELERATING
ASSETS

1000+ MD/Y

SOFTWARE AG INVESTS
IN YOUR SUCCESS

Fully supporting the
DevOps paradigm

PRIME
DevOps

FREE
FOR ALL CUSTOMERS

PRIME
Business & IT Transformation
PRIME
In-Memory Data Management
PRIME
Analytics & Decision
Management
PRIME
Integration Management
PRIME
Process Management
PRIME
Cross Platforms

Aligned with the Digital
Business Platform

Your  **TECHcommunity** account grants you full access to PRIME

Software AG Training | 19 - 2

Notes:

What Should I Take Next?

Now that you have completed the “API Management with webMethods Platform” course, there are other courses that may be interesting for you:

- SOA Governance with CentraSite

softwareag.com/education

Software AG Training | 19 - 3

Notes:



Certification

Our certification programs establish standards for knowledge and skills necessary to successfully implement mission-critical IT systems using Software AG technology.

- webMethods Certified API Management Professional

softwareag.com/education

Software AG Training | 19 - 4

Notes:



Further Information

- Find our Developer Communities at
<http://techcommunity.softwareag.com>

- Contact our Support web site at
<http://empower.softwareag.com>

- Submit your product and feature ideas at Brainstorm
(available within Empower)
<http://empower.softwareag.com>

Software AG Training | 19 - 5

Notes:

Feedback

- Questions and Comments
- Please complete a course evaluation, ...
 - to support us matching your needs
 - to get your certificate



Notes:



Thank You!

Notes:

This page intentionally left blank.

Software AG
Internal Use Only!