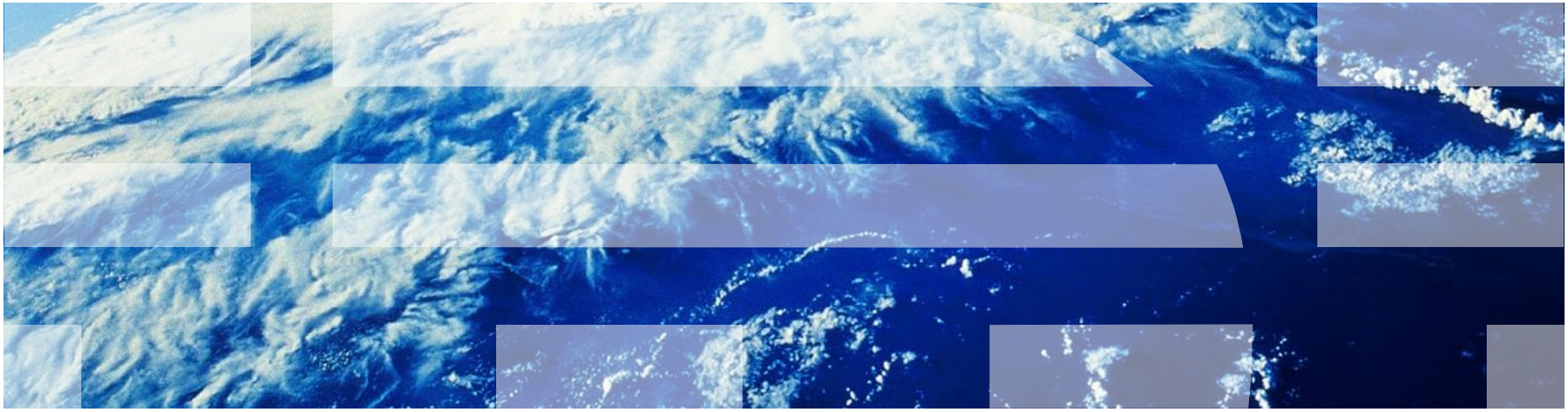
Gautam Kamat & Mangesh Patankar

January 28, 2022

IBM DataPower Gateway

Overview, DevOps



Agenda

- **Overview**
- Security and Integration Gateway
- Application Optimization
- API-Management
- DevOps

WebSphere DataPower Gateway Appliances

SECURE your Mobile, Web, API, B2B and Cloud Workloads

SIMPLIFY your connectivity infrastructure

ACCELERATE your time to value

GOVERN your evolving IT architecture

WebSphere DataPower Appliances provide a low startup cost, helping clients **increase ROI** and **reduce TCO** with specialized, consumable, dedicated appliances that combine **superior performance** and **hardened security** in a variety of form factors

DataPower Gateways

Over 17 years of innovation & over **2,000** global installations

Government

- Agencies and ministries
- Defense and security organizations
- Crown corporations

Banking

- Majority of the big US and European banks
- All of the big 5 Canadian banks
- Numerous regional banks and credit unions

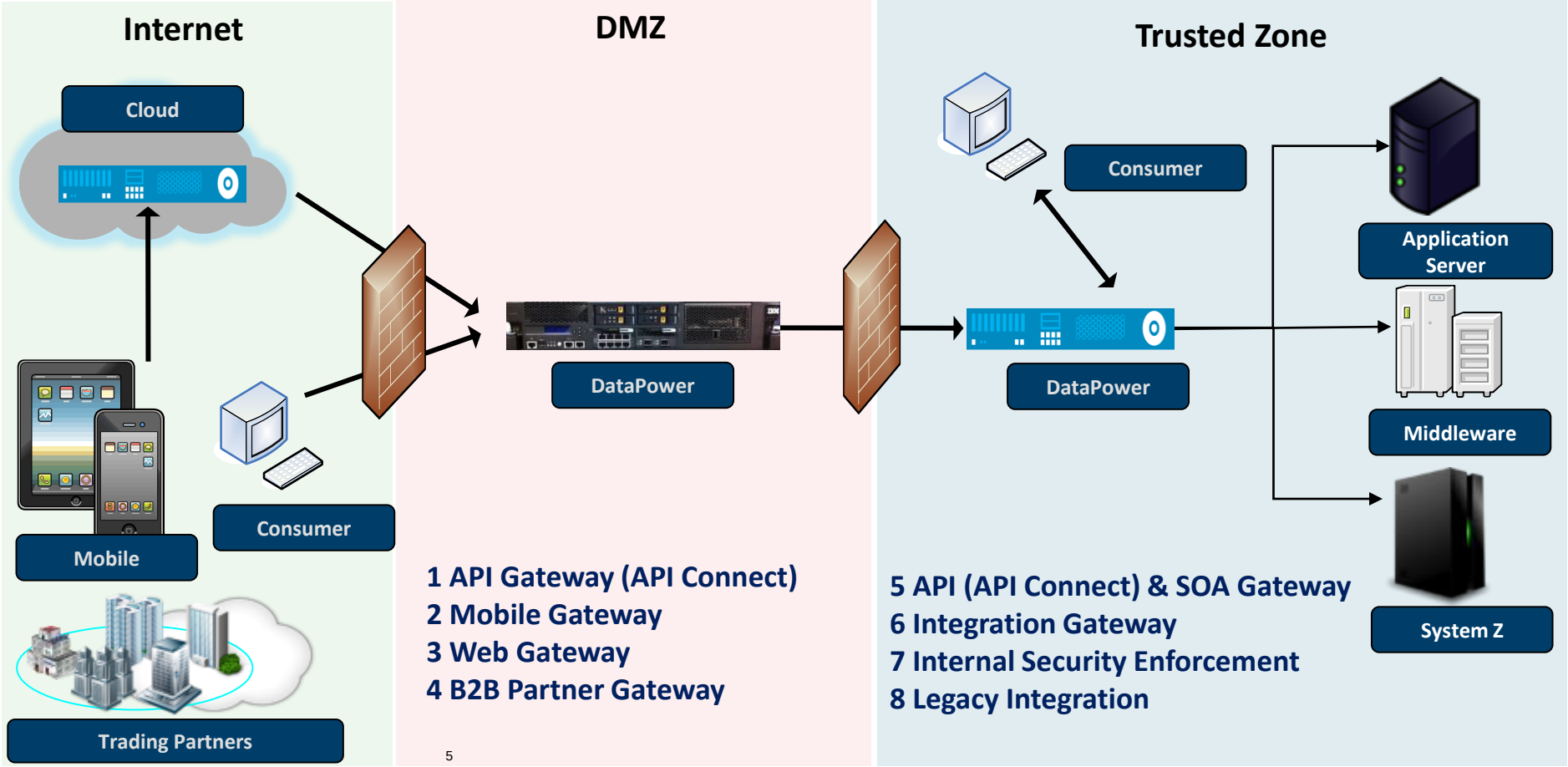
Insurance

- Used by 95% of top global insurances firms
- SaaS providers, ASPs, regulators, etc.

Many, many, more

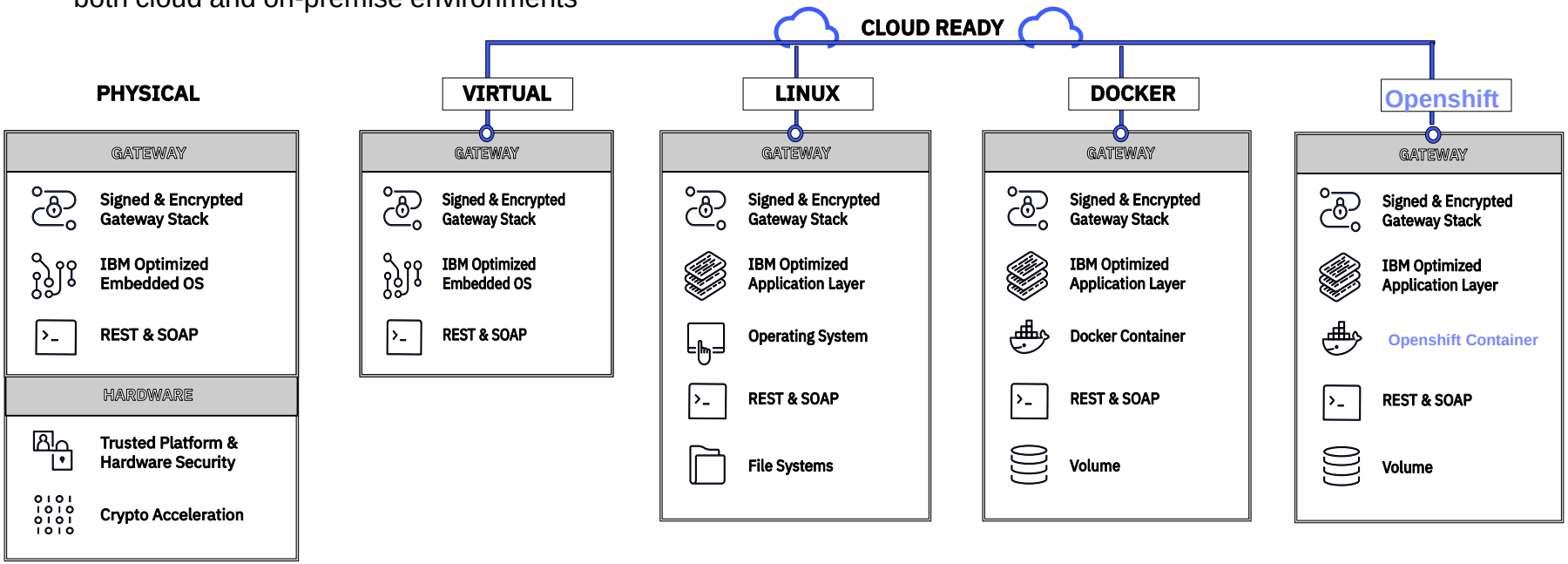
- Healthcare
- Retailers
- Utilities, Power, Oil and Gas
- Telecom
- Airlines
- Others

Industry-leading security & integration gateway



DataPower Gateways can deploy anywhere…

- **Physical appliances**: All-in-one (HW / SW), DMZ-ready with physical security including crypto acceleration and optional hardware security module (HSM)
- **Software**: *virtual* appliance, application (*Linux*) & container (*Docker*) provide flexible deployment options for both cloud and on-premise environments



Key Security and Integration Issues

Secure

- How to **protect** your back-end systems from harmful workloads and unauthorized users?

Control

- How to **shape** traffic based on service level agreements, and **route** based on message content?

Integrate

- How to **convert** payloads, bridge transports and **connect** to existing services at wire-speed?

Optimize

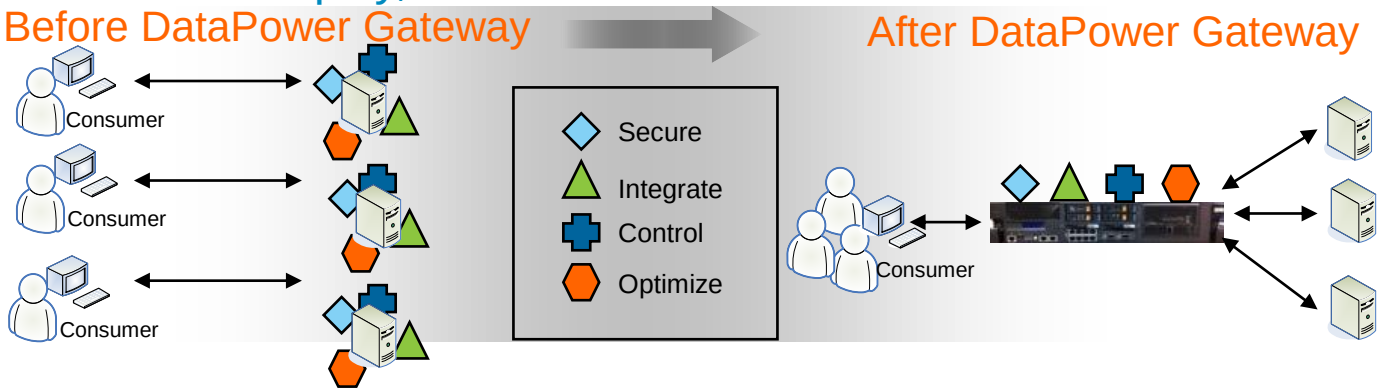
- How to **improve** response time and intelligently **distribute** load?

Core Features

IBM DataPower Gateways

Secure	Integrate	Control	Optimize
Authentication, authorization, auditing	Any-to-any message transformation	Service level management	SSL / TLS offload
Security token translation	Transport protocol bridging	Quota enforcement, rate limiting	Hardware accelerated crypto operations
Threat protection	Message enrichment	Message accounting	JSON, XML offload
Schema validation	Database connectivity	Content-based routing	JavaScript, JSONiq, XSLT, XQuery Response caching
Message filtering & semantics validation	Mainframe connectivity	Failure re-routing	Intelligent load distribution
Message digital signature	B2B trading partner connectivity	Integration with management & visibility platforms	
Message encryption			

Simplify, offload & centralize critical functions

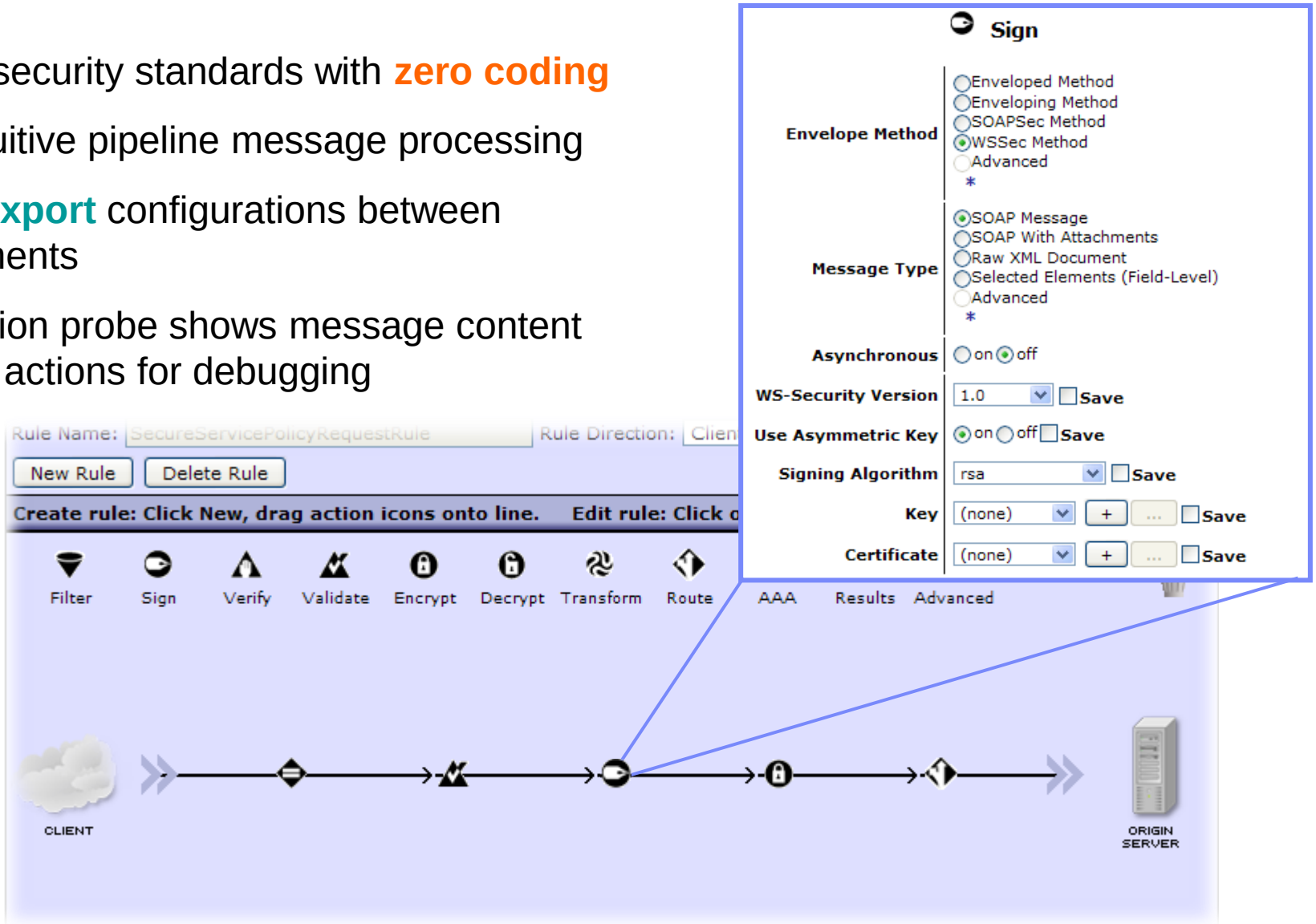


DataPower Gateway: Supported Standards and Protocols

- **Security policy enforcement**
  - OAuth 2.0, OpenID Connect, Social Login
  - JWE, JWS, JWT, JWK
  - SAML 1.0, 1.1 and 2.0, SAML Token Profile, SAML queries
  - XACML 2.0
  - Kerberos (including S4U2Self, S4U2Proxy)
  - SPNEGO
  - RADIUS
  - RSA SecurID OTP using RADIUS
  - LDAP versions 2 and 3
  - Lightweight Third-Party Authentication
  - Microsoft Active Directory
  - FIPS 140-2 Level 3 (w/ optional HSM)
  - FIPS 140-2 Level 1 (w/ certified crypto module)
  - SAF & IBM RACF® integration with z/OS
  - Internet Content Adaptation Protocol
  - W3C XML Encryption
  - W3C XML Signature
  - S/MIME encryption and digital signature
  - WS-Security 1.0, 1.1
  - WS-I Basic Security Profile 1.0, 1.1
  - WS-SecurityPolicy
  - WS-SecureConversation 1.3
- **Data format & language**
  - JavaScript
  - JSON
  - JSON Schema
  - JSONiq
  - REST
  - SOAP 1.1, 1.2
  - WSDL 1.1
  - XML 1.0
  - XML Schema 1.0
  - XPath 1.0
  - XPath 2.0 (XQuery only)
  - XSLT 1.0
- **Transport & connectivity**
  - HTTP, HTTPS, WebSocket Proxy
  - FTP, FTPS, SFTP
  - WebSphere MQ
  - WebSphere MQ File Transfer Edition
  - TIBCO EMS
  - WebSphere Java Message Service
  - IBM IMS Connect, & IMS Callout
  - NFS
  - AS1, AS2, AS3, ebMS 2.0, CPPA 2.0, POP, SMTP (B2B Module)
  - DB2, Microsoft SQL Server, Oracle, Sybase, IMS
- **Transport Layer Security**
  - TLS versions 1.0, 1.1, and 1.2
  - SSL versions 2 and 3
  - SNI, PFS, ECC Ciphers
- **Public key infrastructure (PKI)**
  - RSA, 3DES, DES, AES, SHA, X.509, CRLs, OCSP
  - PKCS#1, PKCS#5, PKCS#7, PKCS#8, PKCS#10, PKCS#12
  - XKMS for integration with Tivoli Security Policy Manager (TSPM)
- **Management**
  - Simple Network Management Protocol
  - SYSLOG
  - IPv4, IPv6
- **Open File Formats**
  - Distributed Management Task Force (DMTF) Open Virtualization Format (OVF)
  - Virtual Machine Disk Format (VMDK)
  - Virtual Hard Disk (VHD)
- **Web services**
  - WS-I Basic Profile 1.0, 1.1
  - WS-I Simple SOAP Basic Profile
  - WS-Policy Framework
  - WS-Policy 1.2, 1.5
  - WS-Trust 1.3
  - WS-Addressing
  - WS-Enumeration
  - WS-Eventing
  - WS-Notification
  - Web Services Distributed Management
  - WS-Management
  - WS-I Attachments Profile
  - SOAP Attachment Feature 1.2
  - SOAP with Attachments (SwA)
  - Direct Internet Message Encapsulation
  - Multipurpose Internet Mail Extensions
  - XML-binary Optimized Packaging (XOP)
  - Message Transmission Optimization Mechanism (MTOM)
  - WS-MediationPolicy (IBM standard)
  - Universal Description, Discovery, and Integration (UDDI versions 2 and 3), UDDI version 3 subscription
  - WebSphere Service Registry and Repository (WSRR)

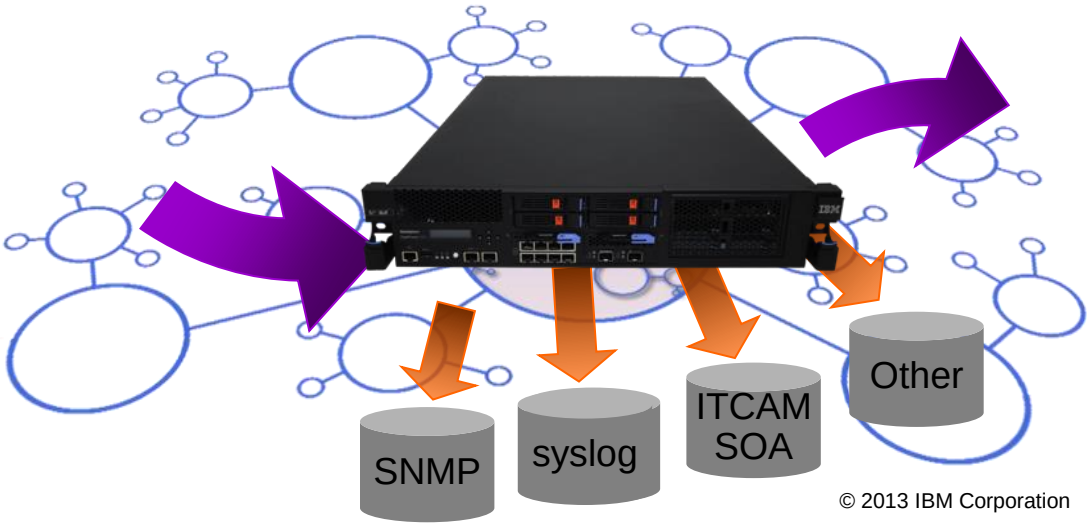
Configuration-driven approach speeds time to market

- Enforce security standards with **zero coding**
- Uses intuitive pipeline message processing
- **Import/export** configurations between environments
- Transaction probe shows message content between actions for debugging



DataPower integrates with current monitoring solutions

- Easily integrate DataPower with your **existing monitoring infrastructure**
  - Extensive health and welfare information via SNMP
  - Detailed transactional information via syslog
- Leverage advanced SOA monitoring tools for more holistic analysis
  - **IBM Tivoli Composite Application Manager (ITCAM) for SOA**
- Create advanced log and audit solutions that meet your application requirements
  - Synchronous or asynchronous logging
  - Guaranteed or best-effort logging
- Customize your monitoring with a flexible log subscription engine
  - Send to **multiple targets**
  - Send in **multiple formats**



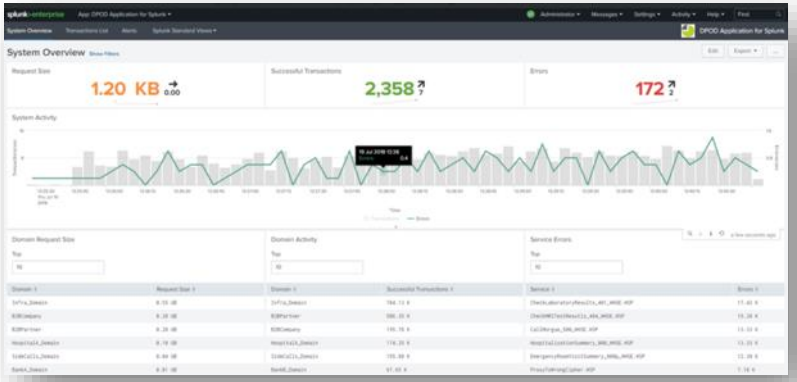
Enhanced Troubleshooting with DataPower Operations Dashboard

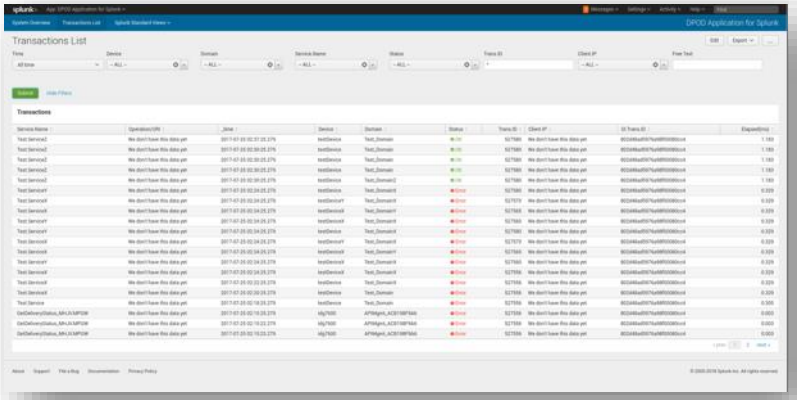
Powerful API diagnostics with detailed views across latency, version, policy, and consumer

Non-intrusive tracking of transactions across multiple gateways without any manual policy instrumentation

Supercharged performance for demanding workloads via new distributed, federated server architecture

Reduce Splunk licensing cost with DPOD plug-in for Splunk, empowering Splunk admins unique operational insight collected from DPOD





Protect your data with cryptography and XML threat protection

- Use DataPower to help resolve **PCI compliance** issues
- Easily sign, verify, encrypt, decrypt any content
- **Configurable** XML Encryption and Digital Signatures
  - Message-level
  - Field-level
  - Headers

XML Threat Protection

- Entity Expansion/Recursion Attacks
- Public Key DoS
- XML Flood
- Resource Hijack
- Dictionary Attack
- Replay Attack
- Message/Data Tampering
- Message Snooping
- XPath or SQL Injection
- XML Encapsulation
- XML Virus
- …many others

See: *The (XML) threat is out there…* by Bill Hines
ibm.com/developerWorks

Efficiently leverage your assets with content-based routing

- Dynamically route based on ***any*** message content
  - Attributes such as the originating IP, requested URL, protocol headers, etc.
  - Data within the message such as SOAP Headers, XML, Non-XML content, etc.

- Query **WebSphere Service Registry & Repository** for routing information
  - Or, use simple XML files
  - Databases
  - Web servers

- Deploy changes to your routing policy with no downtime

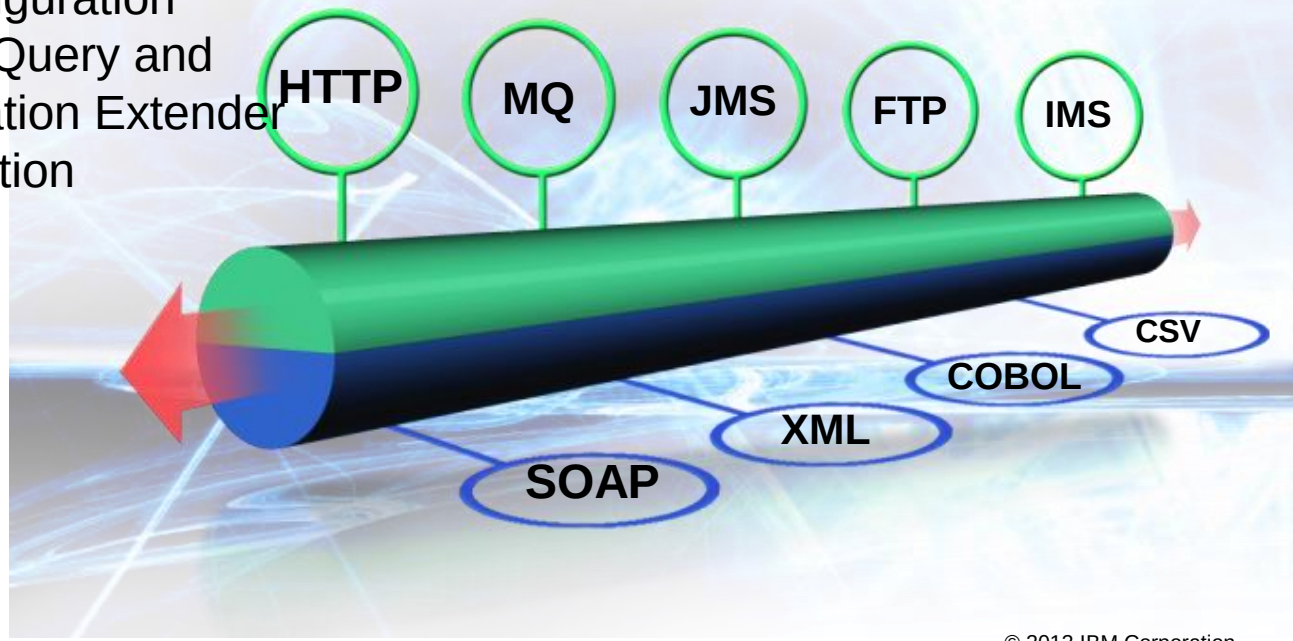
- Convert transport protocol using a simple routing change

Shape traffic with Service Level Management and Load Balancing

- Use **Service Level Management (SLM)** to protect your applications from over-utilization
  - Frequency based on concurrency OR based on messages per time period
  - Take action when exceeding a custom threshold:
    - Notify (or log)
    - Shape (or delay)
    - Throttle (or reject)
- Integration with WebSphere Registry and Repository SLA Policies.
  - Automatically enforce WSRR Policies for runtime governance
  - SLA information cached for efficiency and maximum availability
- Combine SLM with Routing to make intelligent failover decisions
  - Use alternate servers when a threshold is exceeded
- Advanced **Load Balancing** algorithms simplify your architecture
  - First Available
  - (Weighted) Round Robin
  - (Weighted) Least Connections
  - Hash

Bridge across systems with transport and payload transformations

- Integrate disparate **transport protocols** with extreme ease
  - No dependencies between inbound "front-side" and outbound "back-side"
  - Examples: HTTP(s), WebSphere MQ, WebSphere MQ FTE, WebSphere JMS, Tibco EMS, SFTP, FTP(s), NFS, IMS, Database (DB2, Oracle, Sybase, SQL Server, IMS Connect)
- Transform the **message format** with ultimate flexibility
  - Process XML, JSON and other formats in a single configuration
  - Flexibly utilize XSLT, XQuery and WebSphere Transformation Extender for message transformation
- Support synchronous, asynchronous, publish-subscribe and guaranteed-delivery message patterns



Consolidate your infrastructure with Application Optimization

- Use **Self-Balancing** technology to spread inbound traffic load across multiple DataPower appliances using a single target
  - Eliminates the need for additional physical Load Balancers
  - Efficiently distributes traffic with minimal overhead

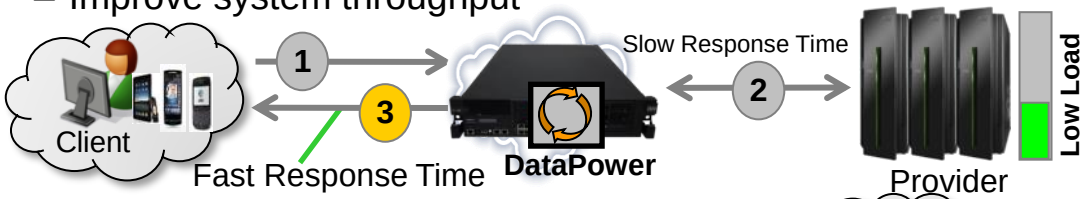
- **Embedded On Demand Router** (ODR) to intelligently proxy HTTP traffic to **WAS ND** environments

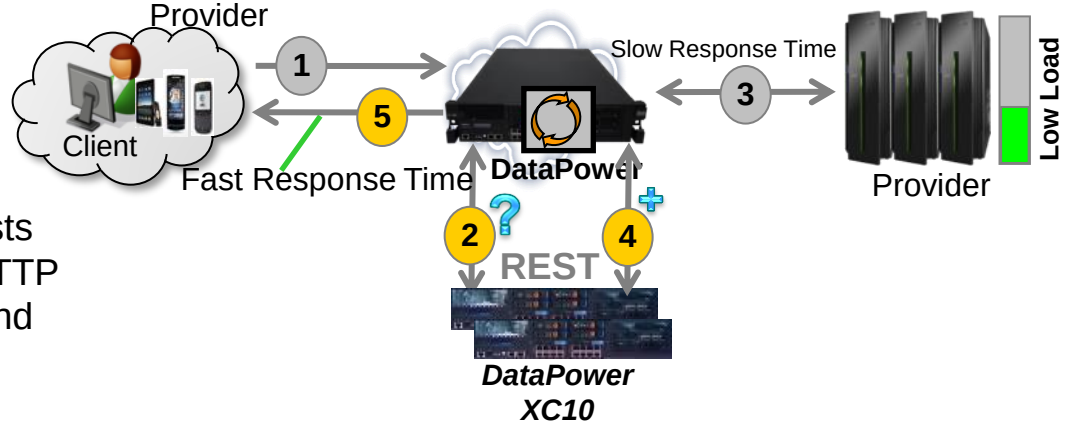
- Use **Session Affinity** to preserve target session state across multiple requests
  - Supports WebSphere and Non-WebSphere targets

Optimization: Backend Response Caching

Accelerate workload delivery & reduce load on backend systems

- **Policy-driven** local '**on-box**' backend response caching & seamless integration with **XC10** appliances for '**off-box**', shared, elastic caching
  - Improve client observed response time
  - Reduce backend server load
  - Improve system throughput





- **Features**
  - Cache HTTP(s) GET, PUT, POST requests
  - Drive caching automatically based on HTTP 1.1 Cache-Control headers set by backend system
  - Return stale documents
  - Smart RESTful cache invalidation
  - Flexibility of utilizing user-defined cache key instead of default URI
  - Little or no XSLT required

Document Cache Policy

URL Match Expression	Policy Type	TTL	Priority	XC10 Grid	Cache Back-end Responses	HTTP Cache Validation	Return Expired Document	RESTful Invalidation	Cache Response to POST and PUT Requests

Enhanced REST Service Workload Processing

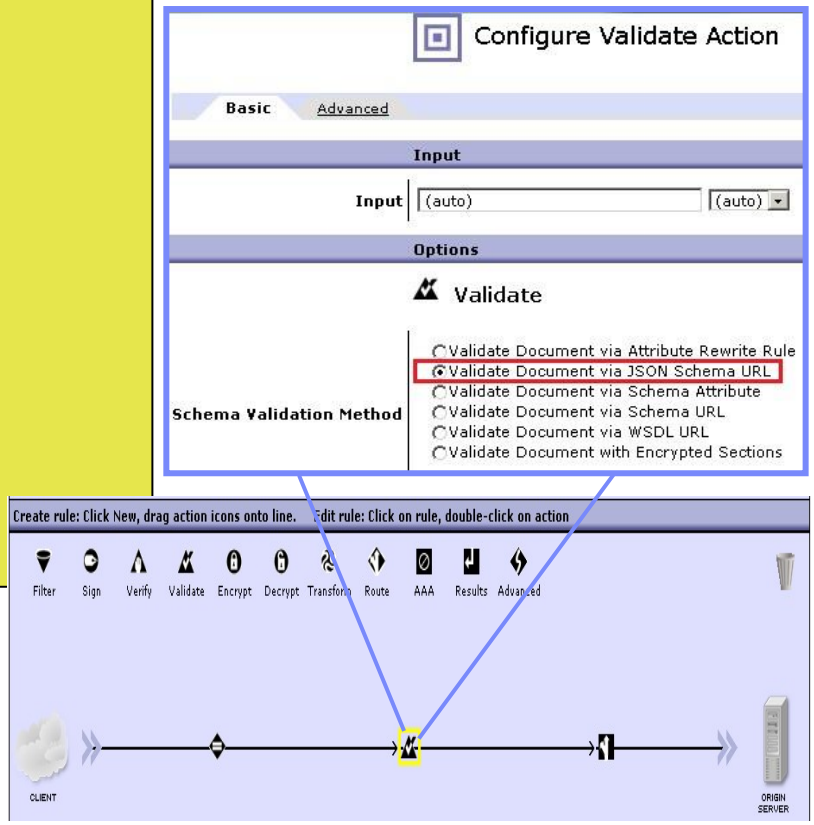
Native JSON support for enhanced security & control of REST services

- Native high speed JSON parsing / transformation at near wire speed
  - High-speed parsing and tuned compilation with native execution
- JSON **schema validation**: Security & input validation
  - Support for draft 3 of IETF specification

```
{ "name"   : "John Smith",
  "sku"    : "20223",
  "price"  : "23.95",
  "shipTo" : { "name" : "Jane Smith",
               "address" : "123 Maple Street",
               "city" : "Pretendville",
               "state" : "NY",
               "zip" : "12345" },
  "billTo" : { "name" : "John Smith",
               "address" : "123 Maple Street",
               "city" : "Pretendville",
               "state" : "NY",
               "zip" : "12345" }
}
```

JSON Schema

```
{
  "type": "object",
  "properties": {
    "name": { "type": "string" },
    "sku": { "type": "string" },
    "price": { "type": "number" },
    "shipTo": {
      "type": "object",
      "properties": {
        "name": { "type": "string" },
        "address": { "type": "string" },
        "city": { "type": "string" },
        "state": { "type": "string" },
        "zip": { "type": "string" }
        }
    },
    "billTo": {
      "type": "object",
      "properties": {
        "name": { "type": "string" },
        "address": { "type": "string" },
        "city": { "type": "string" },
        "state": { "type": "string" },
        "zip": { "type": "string" }
        }
    }
  }
}
```



Enhanced REST Service Workload Processing

Native JSON support for enhanced security & control of REST services

- Query, extract, filter, transform of JSON messages using **JSONiq**
  - Extension to XQuery: Like SQL for JSON and XML

```
[{ "given" : "John",  "surname" : "Smith", "sku" : "20223", "price" :  23.95},
 { "given" : "Alice", "surname" : "Brown", "sku" : "54321", "price" : 199.95},
 { "given" : "John",  "surname" : "Smith", "sku" : "23420", "price" : 104.95},
 { "given" : "Bob",   "surname" : "Green", "sku" : "90231", "price" : 300.00},
 { "given" : "Scott", "surname" : "Jones", "sku" : "54321", "price" : 199.95},
 { "given" : "Jim",   "surname" : "Lee",   "sku" : "89820", "price" :  46.50}]
```

Query

```
declare option jsoniq-version "0.4.42";
for $x in jn:members(.)
  where $x("price") >= 100.00
  order by $x("surname")
  return concat($x("given"), ' ', $x("surname"), '&#xA;')
```

```
Alice Brown
Bob Green
Scott Jones
John Smith
```

```
{ "name"   : "John Smith",

  "sku"    : "20223",

  "price"  : "23.95",

  "shipTo" : { "name" : "Jane Smith",

               "address" : "123 Maple Street",

               "city" : "Pretendville",

               "state" : "NY",

               "zip" : "12345" },

  "billTo" : { "name" : "John Smith",

               "address" : "123 Maple Street",

               "city" : "Pretendville",

               "state" : "NY",

               "zip" : "12345" }

}
```

Extract

```
declare namespace output = "http://www.w3.org/2010/xslt-xquery-serialization";
declare option jsoniq-version "0.4.42";
declare option output:method "json";
.("shipTo")
```

```
{ "name"    : "Jane Smith",

  "address" : "123 Maple Street",

  "city"    : "Pretendville",

  "state"   : "NY",

  "zip"     : "12345"

}
```

Filter

```
declare namespace output =
 "http://www.w3.org/2010/xslt-xquery-serialization";
declare option jsoniq-version "0.4.42";
declare option output:method "json";

if (.("shipTo")("state") = "HI")
then fn:error(fn:QName('http://example.org/mine',
            'myerr:noshipHI'),
            'Sorry, we do not ship to Hawaii.')
```

```
*** ABORTED: Error noshipHI: Sorry, we do not ship to Hawaii.
```

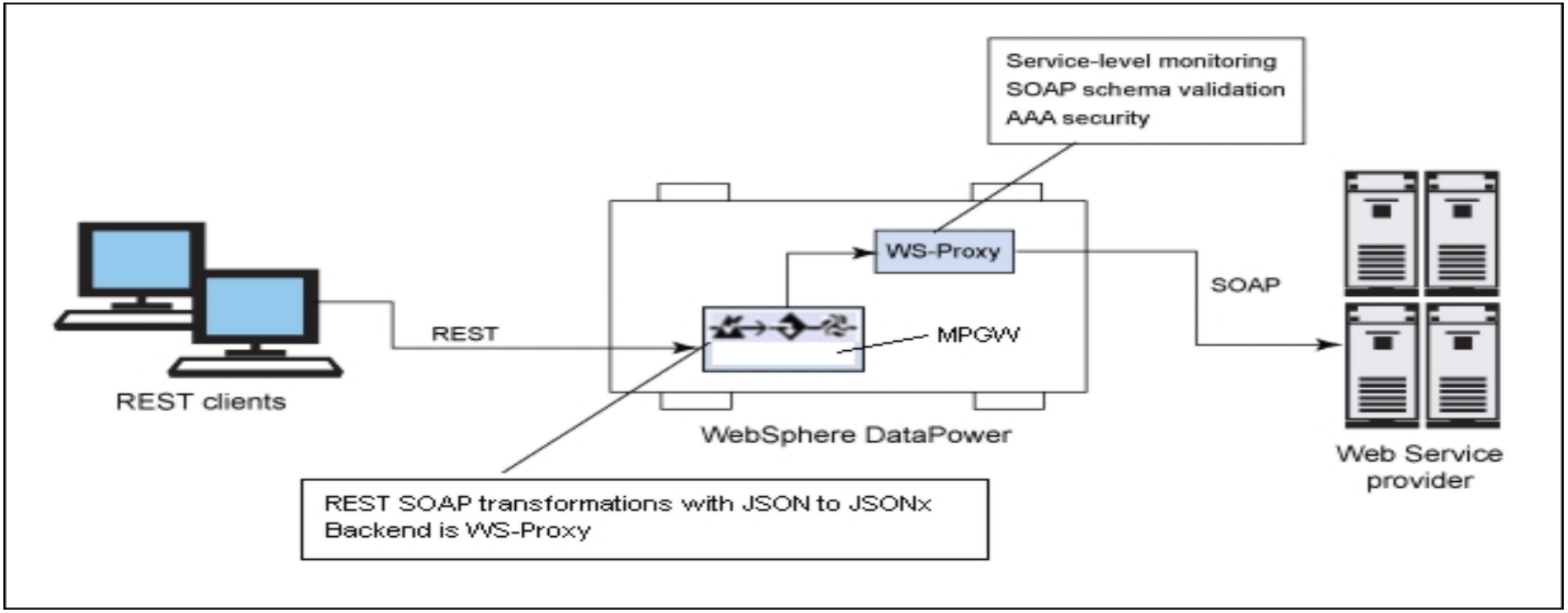
Transform

```
declare option jsoniq-version "0.4.42";
<order>
  <name>{.("name")}</name>
  <price>{.("price")}</price>
  <state>{.("shipTo")("state")}</state>
</order>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<order><name>John Smith</name><price>23.95</price><state>NY</state></order>
```

Expose legacy SOAP Services as REST with DataPower

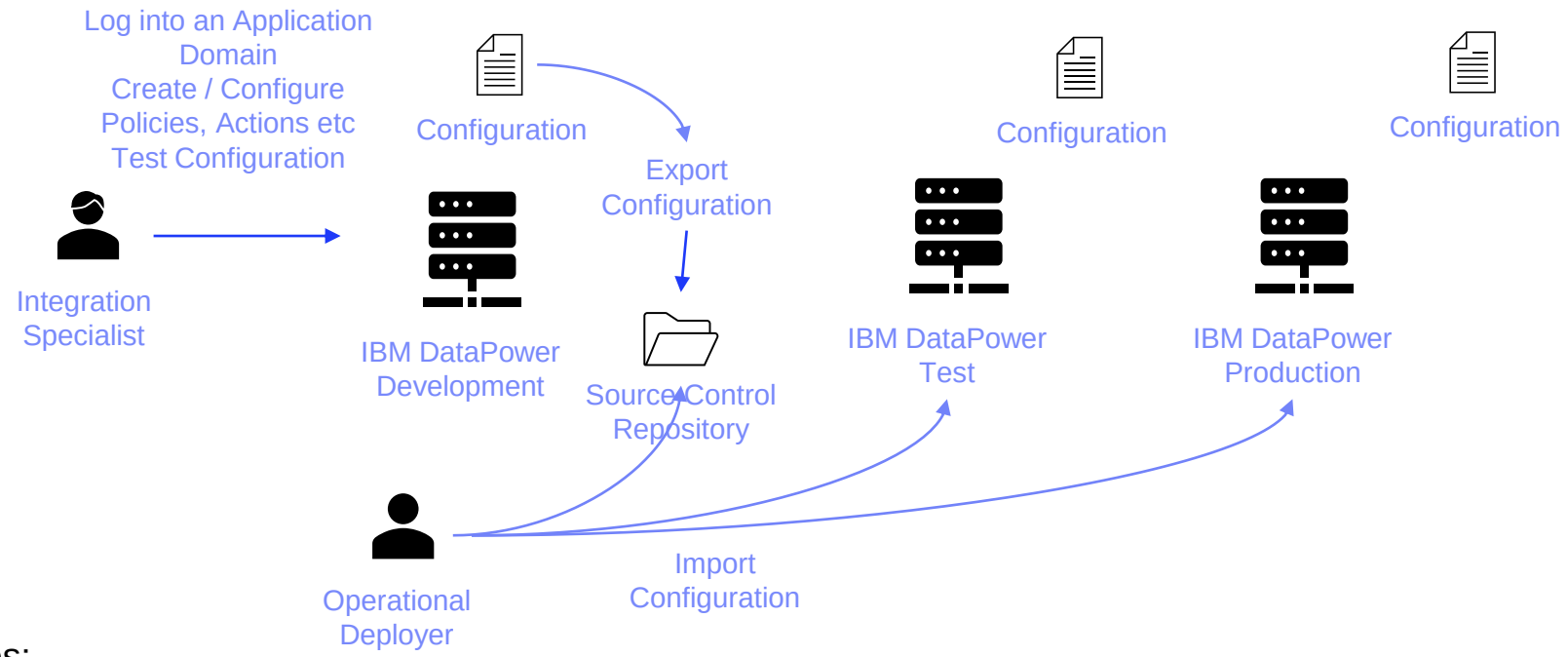
- DataPower transforms JSON to/from SOAP/XML
- DataPower bridges mobile side OAuth, HTTP Basic Auth to SOAP Based WS-Security, SAML or legacy authentication/authorization security methods
- DataPower exposes both original SOAP service for traditional clients and a REST/JSON front end for Mobile clients



DevOps – What is ?

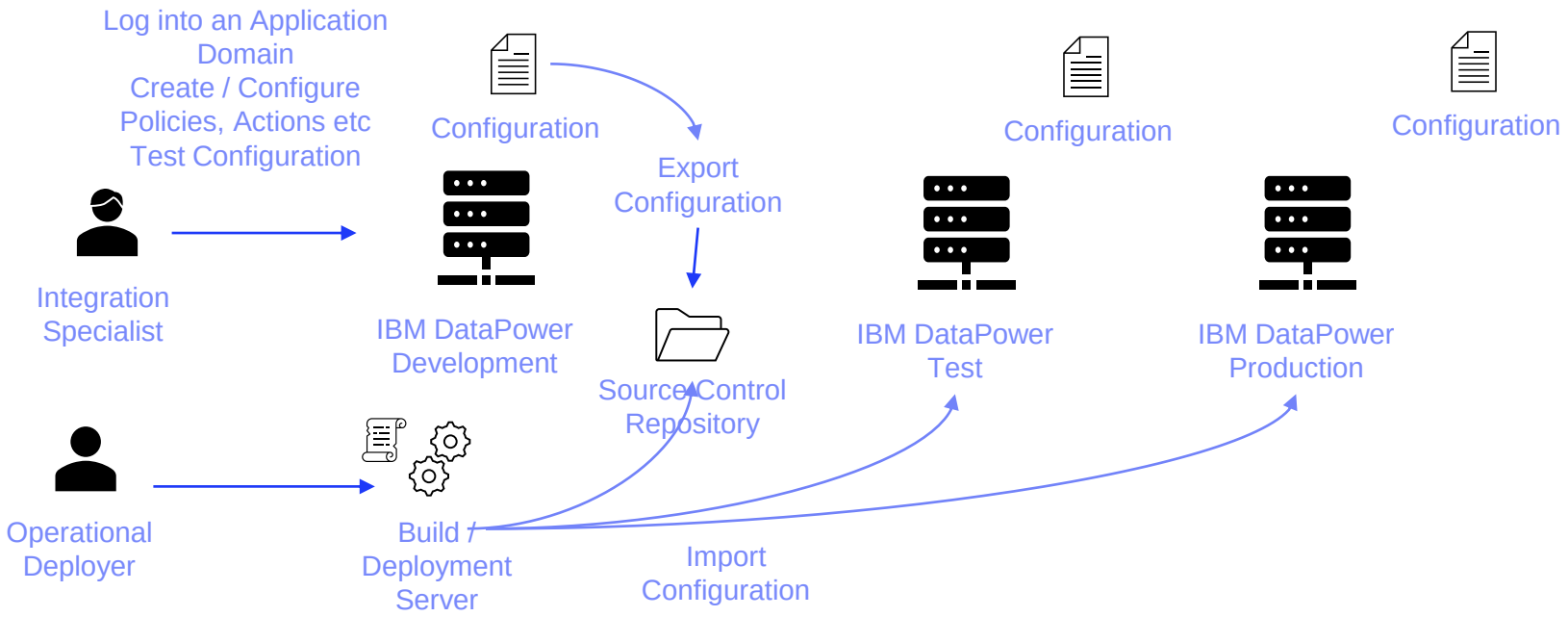
- A way or a method for developing and releasing software
- Collaborative efforts between development and deployment to enable shorter iterations, faster releases
- Involves automation by stringing together different tools
- Pipeline
  - Code is committed to a source control repository
  - CI / CD triggers
  - Code is compiled, packaged, configured
  - Unit tests are run
  - Integration tests are run
  - Code is delivered and deployed
- Pipeline design
  - Identify manual steps, sequence of activities, repetitive tasks
  - Process frequency, length, dependencies, impact
  - Tooling requirements
  - Target environment (on-prem vs cloud)
  - Handling build / deployment failures
    - Email / SMS notifications
    - Build logs and reports
- Automation tools:
  - Red Hat Ansible
  - Jenkins
  - Chef
  - Git
  - Nexus
  - Artefactory
  - Maven
  - UrbanCode Deploy
  - JMeter

DevOps – Manual Approach



- Pros:
  - Simple and easy
  - Use built-in DP features
  - All activities within DP RBAC
- Cons:
  - Not scalable
  - Not repeatable / consistent
  - Operations Team – specialist role
  - Configurations in each domain can get inconsistent over time.

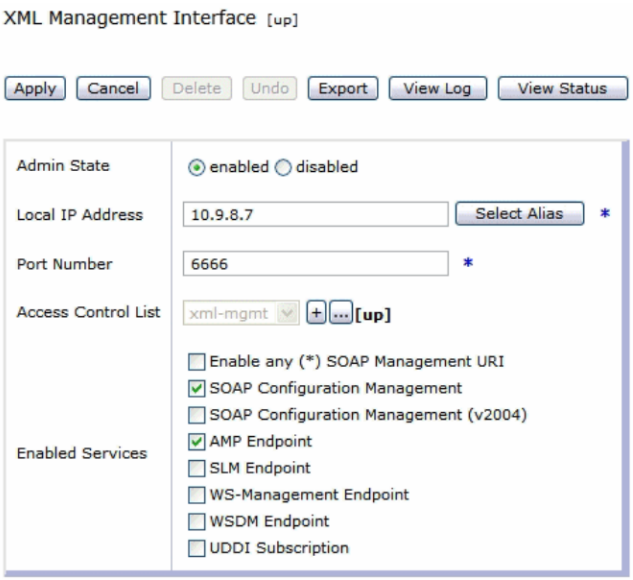
DevOps – Roll-Your-Own Approach (XMI and RMI)



- Pros:
  - Simple and easy
  - Use one of DP management interfaces
    - XML Management Interface, REST Management Interface, Command Line Interface
  - All activities within DP RBAC
  - Repeatable / consistent
- Cons:
  - Automation via scripts
  - Scripts can get inconsistent, need more maintenance
  - Skills needed to create and maintain scripts
  - Configurations in each domain can get inconsistent over time.

DevOps – XML Management Interface

- Can be enabled from the Web GUI and the CLI
- Use simple curl commands
  - $ curl -k -u user:password -d @request.xml http://datapower-host:port/endpoint-uri
  - Few endpoint URI:
    - SOAP Configuration Management: /service/mgmt/current
    - Appliance Management Protocol: /service/mgmt/amp/1.0
- AMP:
  - store:///app-mgmt-protocol.wsdl
  - store:///app-mgmt-protocol.xsd



- Refer IBM Redbook “WebSphere DataPower SOA Appliance: The XML Management Interface” (redp4446)

DevOps – XML Management Interface

Generic Request Structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2001/12/soap-envelope">
  <env:Body>
    <dp:request xmlns:dp="http://www.datapower.com/schemas/management">
      …
    </dp:request>
  </env:Body>
</env:Envelope>
```

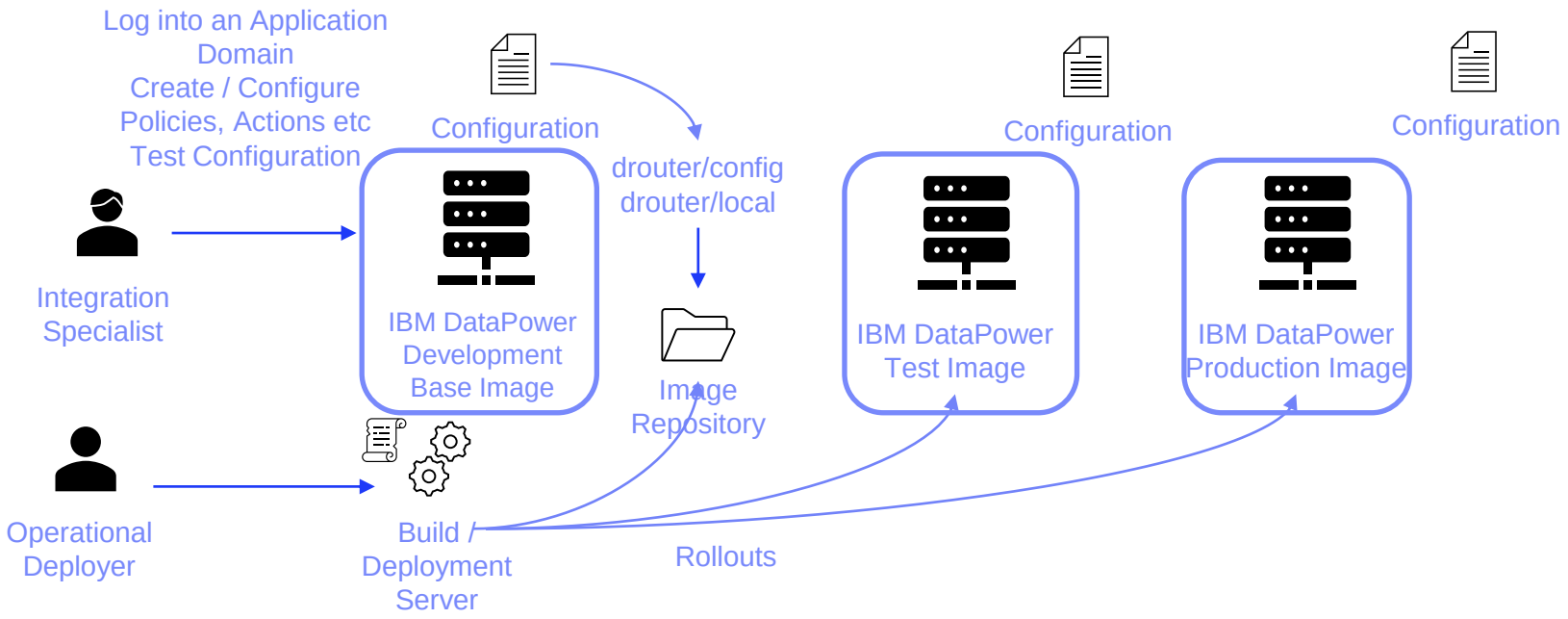
Generic Response Structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Body>
    <dp:response xmlns:dp="http://www.datapower.com/schemas/management">
      <dp:timestamp>timestamp</dp:timestamp>
      …
    </dp:response>
  </env:Body>
</env:Envelope>
```

DevOps – XML Management Interface

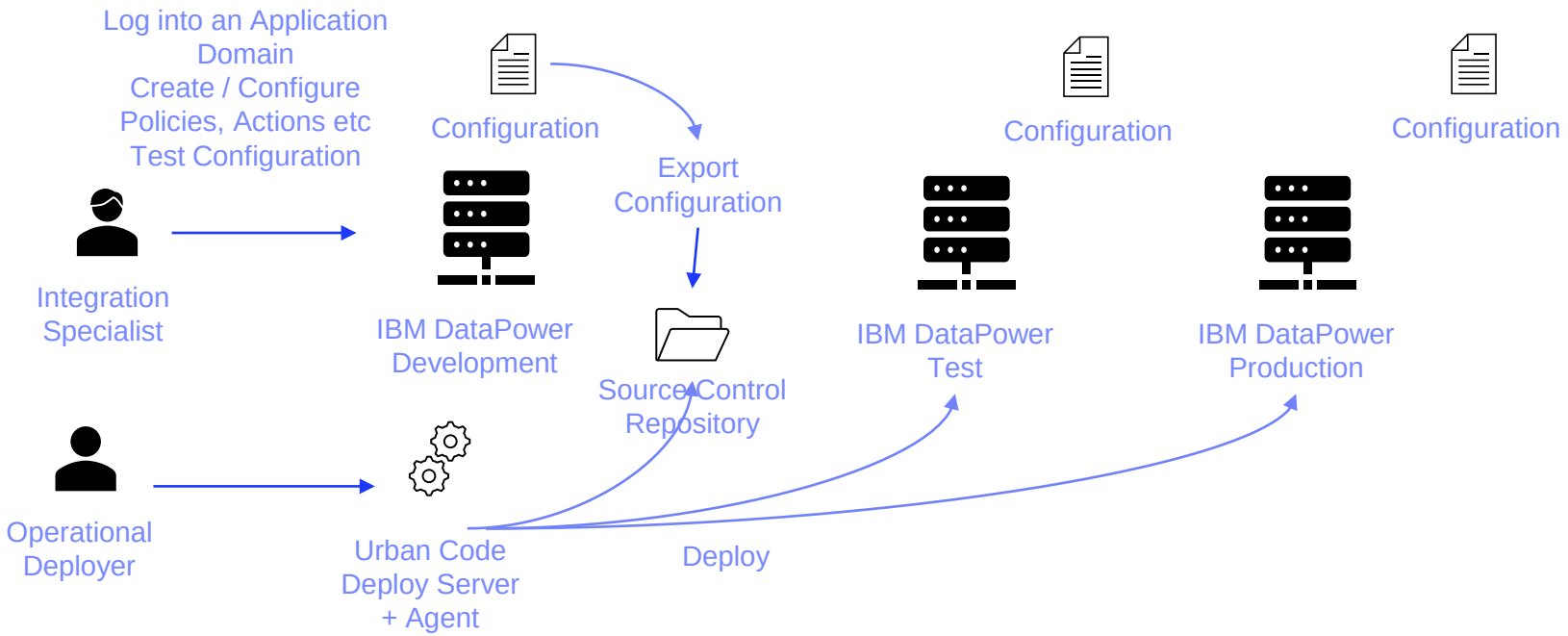
Retrieve login token	<dp:get-samlart>
Retrieve a status object	<dp:get-status>.
Compare configurations	<dp:get-diff>
List files and directories	<dp:get-filestore />
Retrieve log data	<dp:get-log /> <dp:get-log name="logTarget" />
Download a file	<dp:get-file name="directory:///file" />
Upload a file	<dp:set-file name="directory:///file"> *** base64 encoded file *** </dp:set-file>
Export configuration data	<dp:do-export>
Import configuration data	<dp:do-import>
Create a backup of a domain	<dp:do-backup>
Restore a domain from a backup	<dp:do-restore>
Create an object	<dp:set-config>, <dp:get-config class="ConfigEnum" />, <dp:get-config class="ConfigEnum" name="name" />
Modify the configuration of an object	<dp:modify-config>
Delete an object	<dp:del-config>

DevOps – Roll-Your-Own Approach (Docker)



- https://www.ibm.com/docs/en/datapower-gateways/10.0.1?topic=docker-creating-datapower-application

DevOps – UrbanCode Deploy



- Pros:
  - Full application domain, firmware life-cycle management
  - Integration with 3P SCM
  - All activities within DP RBAC
  - Easily understood by operator teams
  - Rollback if needed
- Cons:
  - Need installation (server / agent)
  - Plugin configuration

Thank you !

Backup Slides

DataPower Family

Service Gateway XG45

- Entry-level device, slim footprint (1U)
- Security gateway (AAA, XML threat, etc)
- Service level management and monitoring
- Intelligent load distribution & dynamic routing
- Lightweight integration functions (optional)
- Available in Virtual Edition

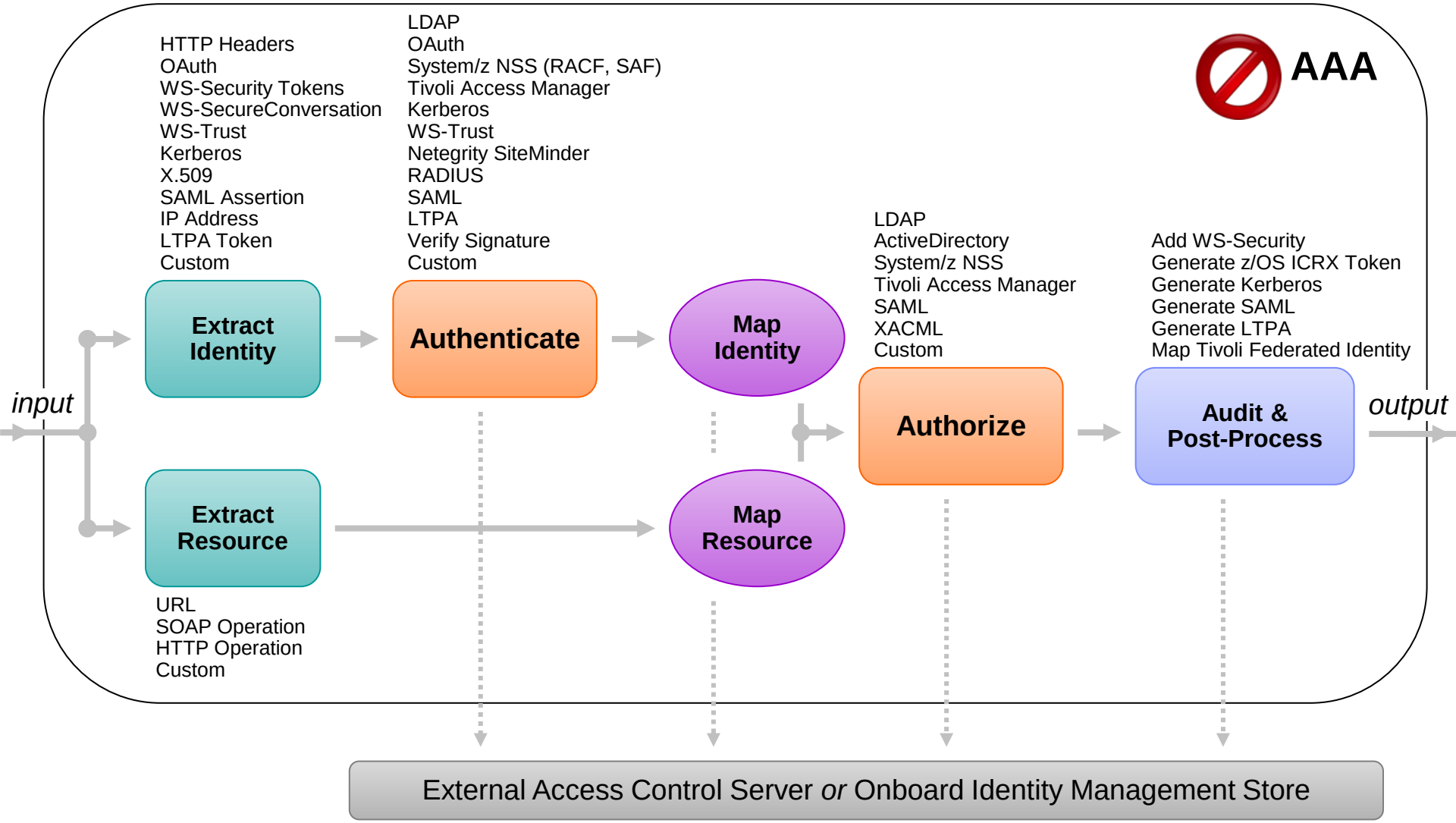
Integration Appliance XI52

- High density 2U form, XG45 functionality plus
- "Any-to-Any" conversion at wire-speed
- Bridges multiple transport protocols
- Mainframe integration & enablement
- Available in Virtual Edition

B2B Appliance XB62

- High density 2U form, XI52 functionality plus
- B2B Messaging (AS1/AS2/AS3/ebMS + CPPA)
- Trading Partner Profile Management

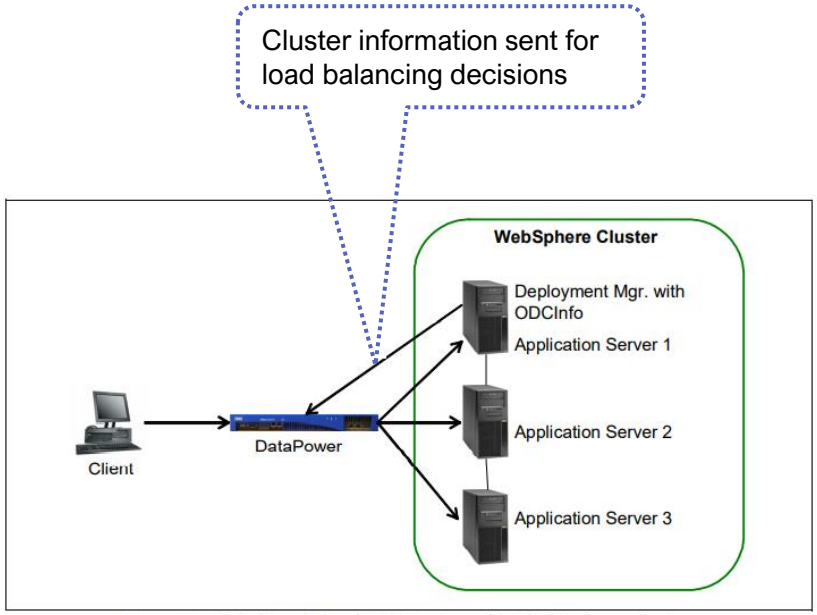
Employ flexible AAA (Authenticate, Authorize, Audit) Policies



DataPower – Load Balancing

Load Balancing – Backend Destination

- Done through Load Balancer Groups
- Load Balance Group defines the group members
- Each member is identified by their Hostname / IP, port, health port, weights
- WebSphere Cell is created to reference the Load Balancer Group to keep it updated with the cluster changes published by the remote server cluster (ODCInfo)
- Load balancing policy, session affinity to use are decided
- An XML Manager configuration is created referencing the Load Balancing Group
- A service is created referencing the XML Manager and the backend URL is configured to use the Load Balancer Group name
- Additional requirements especially around load balancing of non-WebSphere Application Servers



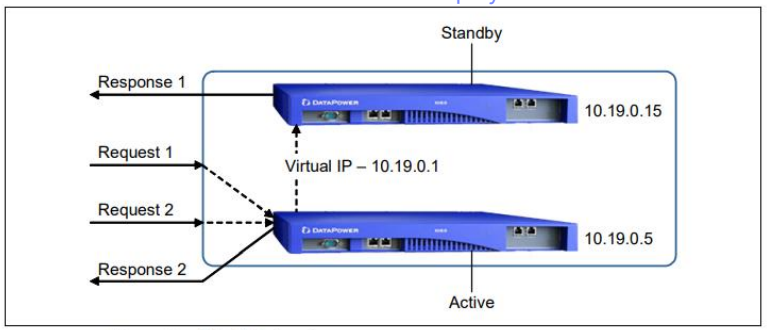
Example of Intelligent Load Distribution of a WebSphere cluster

Redbooks: sg247901
IBM DataPower Administration & Deployment Best Practices

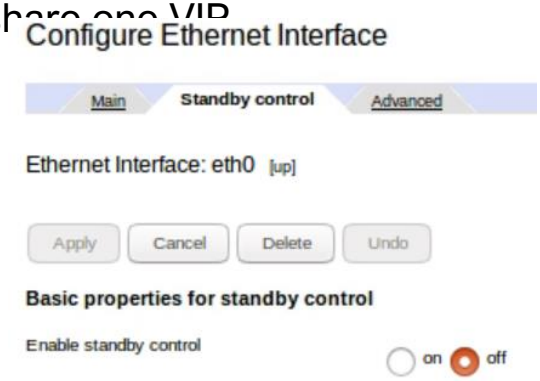
Load Balancing – Self-Balancing

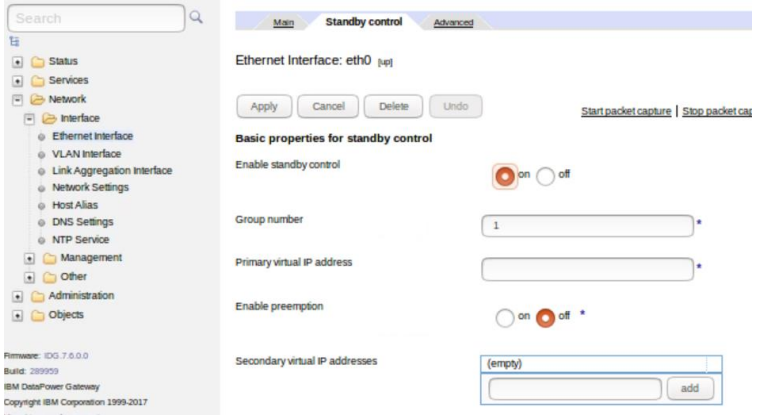
- Two or more DataPower appliances can distribute load amongst themselves
- Removes the requirement for traditional load balancers
- Built on top of Standby Control Configuration responsible for a collection of ethernet interfaces on separate appliances to share one VIP

Redbooks: sg247901
IBM DataPower Administration & Deployment Best Practices



Example of Self-Balancing





Load Balancing – Deployment Platforms

- Appliance and DataPower Gateway for VMware
  - Add instances to a tier of locally load balanced DPGW instances
  - Tier in other data centers with geographic load balancer
- DataPower Gateway for Linux
  - Hypervisor and cloud tools to create new VMs
- DataPower Gateway for Docker
  - Container orchestration tools