How to choose the right Technology, Framework or Tool to Build Microservices

Kai Wähner

kwaehner@tibco.com

@KaiWaehner

www.kai-waehner.de

Xing / LinkedIn → Please connect!

TIBCO™ | Key Messages

- Integration is key for success of Microservices!
- Real time event correlation is the game changer!
- TCO and Time-to-Market are major aspects for tool selection!

TIBCO™ | Agenda

- Definition of a Microservice
- Architecture Requirements
- Concepts for Microservices
- **Frameworks and Tools**
- Getting Started

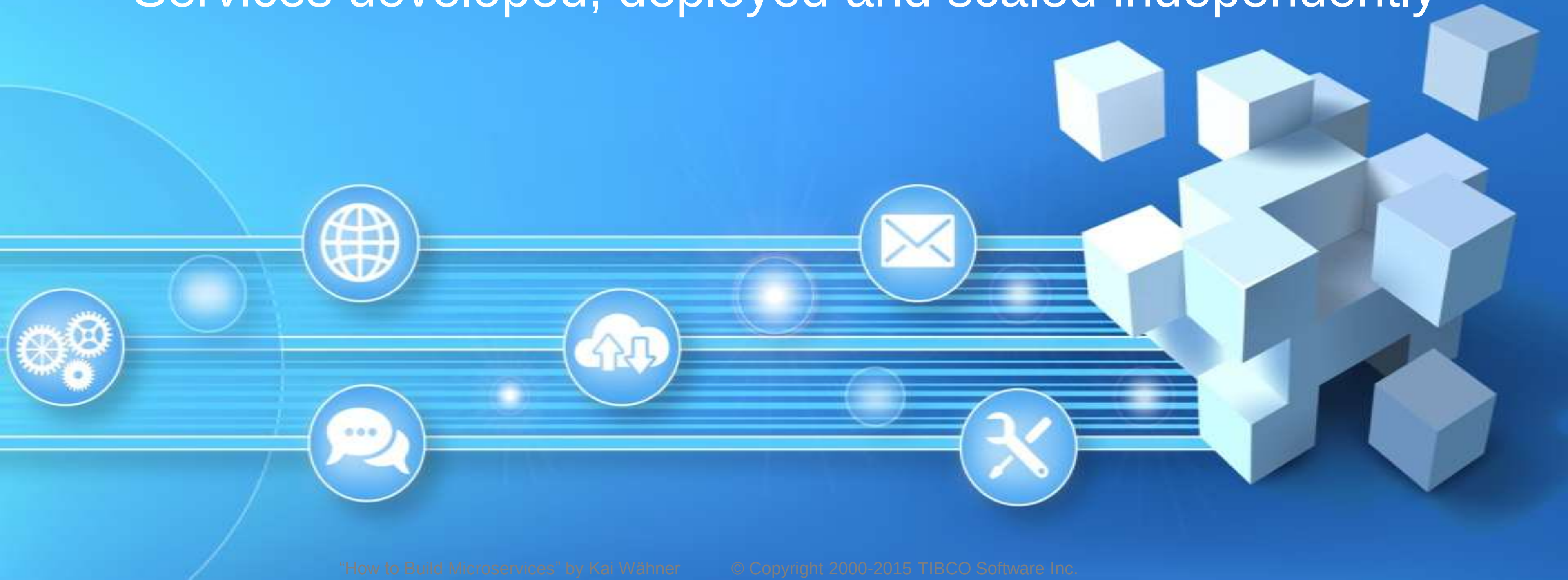
TIBCO™

Agenda

- **Definition of a Microservice**
- Architecture Requirements
- Concepts for Microservices
- Frameworks and Tools
- Getting Started

TIBCO™ | Microservices

- Services implementing a limited set of functions
- Services developed, deployed and scaled independently



Benefits of Microservices

Shorter time to results

- Scale development and reuse of services
- Use the right technology for the job

Increased flexibility

- Change / improve any Microservice without major disruption on apps or other services

Mhhh….

TIBCO™ | Sounds like SOA?

Microservices clearly specify important differences to SOA
(as we see SOA implemented in most enterprises today):

- No commitment to a unique technology ← Avoid a jungle of technologies!
- Greater flexibility of architecture
- Services managed as products, with their own lifecycle
- Industrialized deployment
- Dumb routes and smart endpoints instead of a heavyweight ESB

Integration still needed somewhere!

TIBCO™ | Challenges of Microservices

- All of these services require integration.
- All of these services and technologies require automation of deployment and configuration.
- All of these services require logging and monitoring.
- All of these services require hybrid deployment.

TIBCO™ | Agenda

- Definition of a Microservice
- **Architecture Requirements**
- Concepts for Microservices
- Frameworks and Tools
- Getting Started

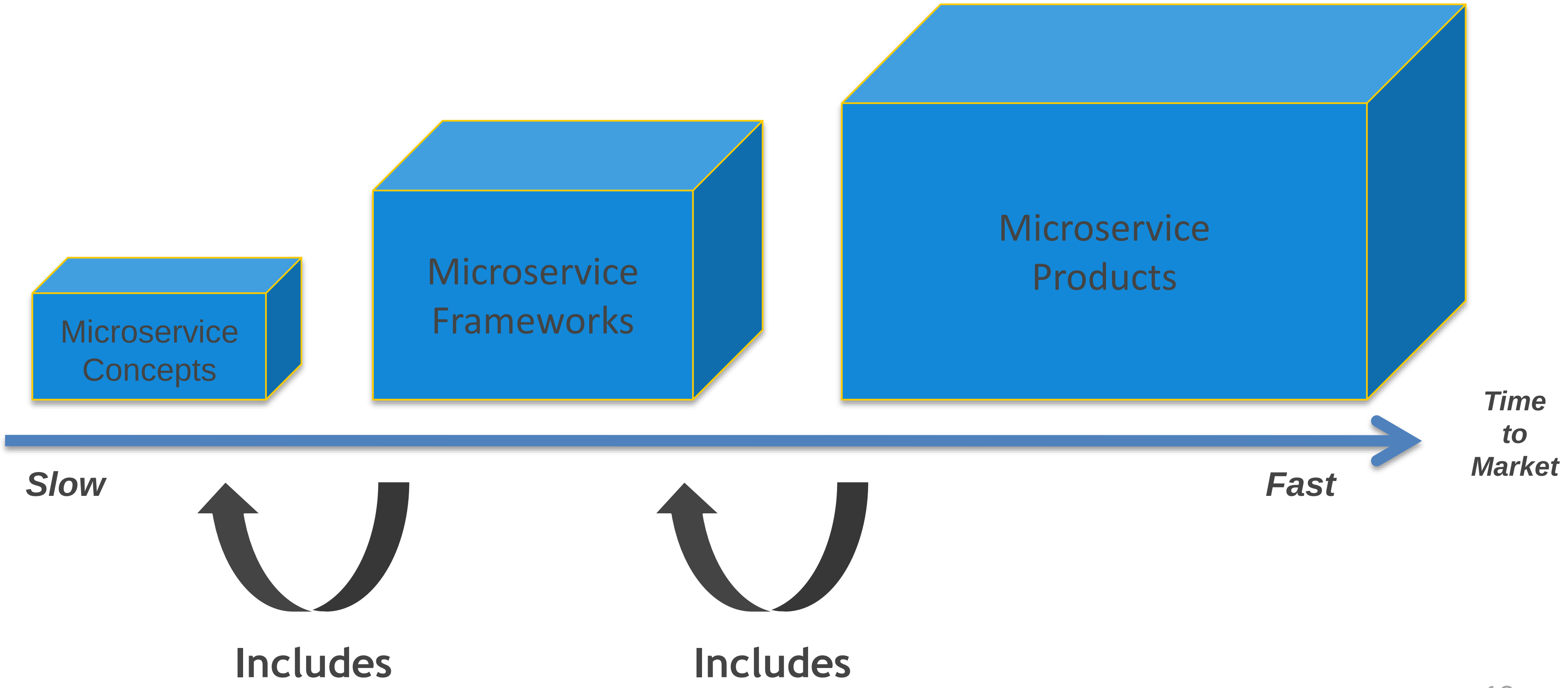
TIBCO™ | Requirements for a Microservices Architecture

① Service Contracts
② Exposing new and existing Services
③ Discovery of Services
④ Coordination Across Services
⑤ Managing Complex Deployments and their Scalability
⑥ Visibility and Correlation across Services

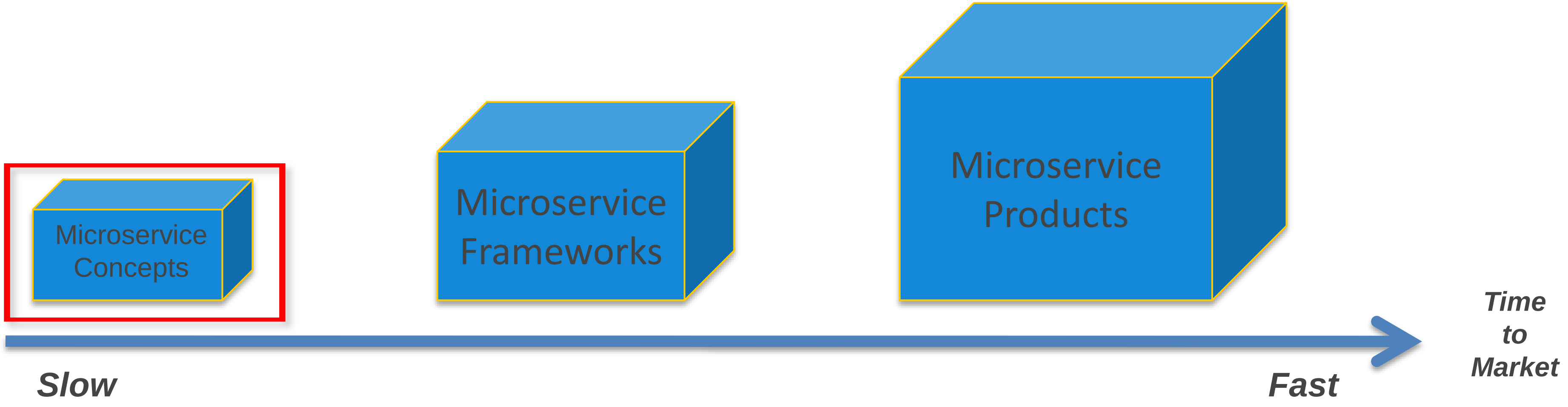
Agenda

- Definition of a Microservice
- Architecture Requirements
- **Concepts for Microservices**
- Frameworks and Tools
- Getting Started

Alternatives to Realize Microservices



What Microservice Concepts do you need?

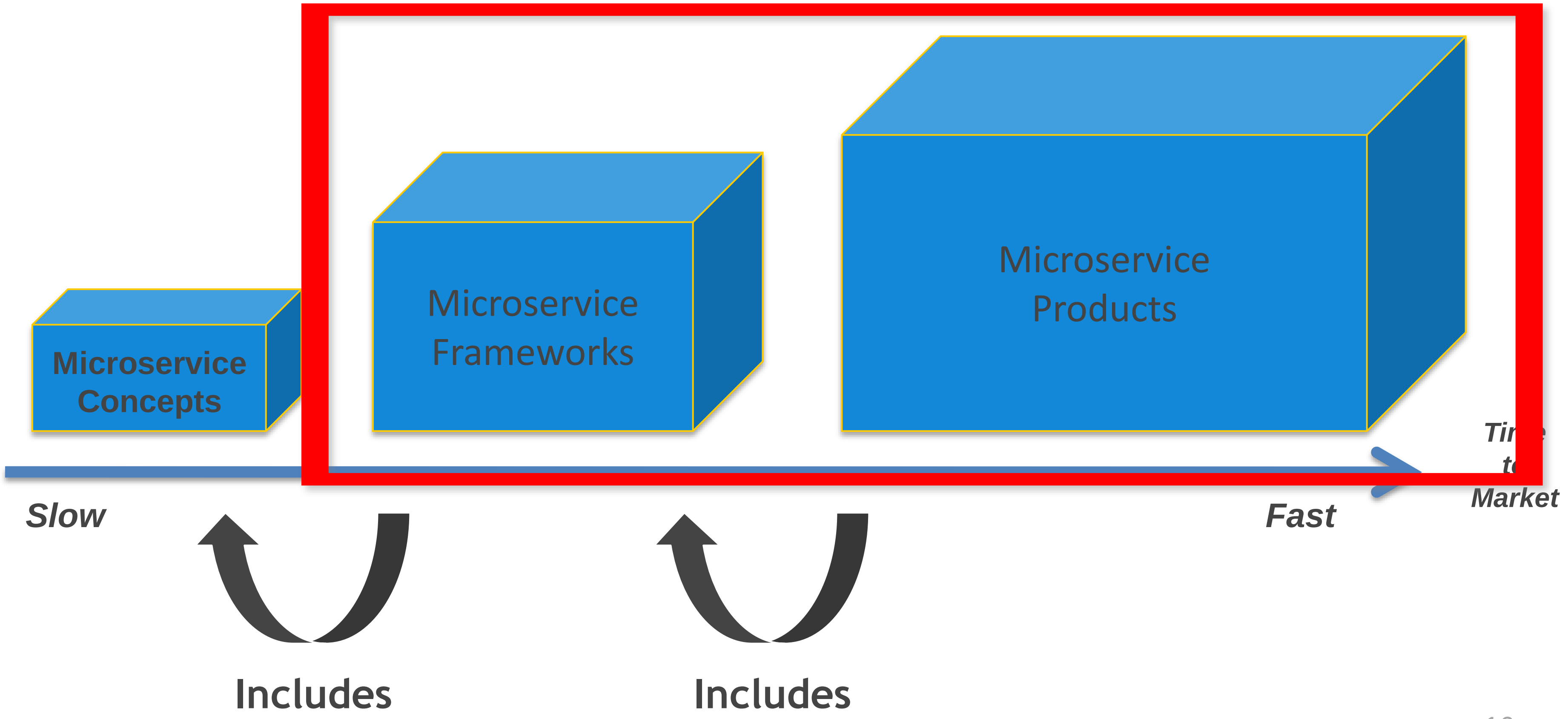


Service Standards (OSGi, REST, SOAP)
Messaging Standards (JMS, MQTT, AMQP, WebSockets)
Continuous Delivery (Build, Deployment) and Elasticity
Service Management (Registry, Governance, Documentation)
Centralized Monitoring and Analytics (Log Analytics, Event Processing)

TIBCO™ | Agenda

- Definition of a Microservice
- Architecture Requirements
- Concepts for Microservices
- **Frameworks and Tools**
- Getting Started

Alternatives to Realize Microservices



TIBCO™

Requirements for a Microservices Architecture

① Service Contracts
② Exposing new and existing Services
③ Discovery of Services
④ Coordination Across Services
⑤ Managing Complex Deployments and their Scalability
⑥ Visibility and Correlation across Services

Requirements for Microservices Architecture

#1: Services Contract

Service Contracts

Service provider express the purpose of the Microservice, and its requirements

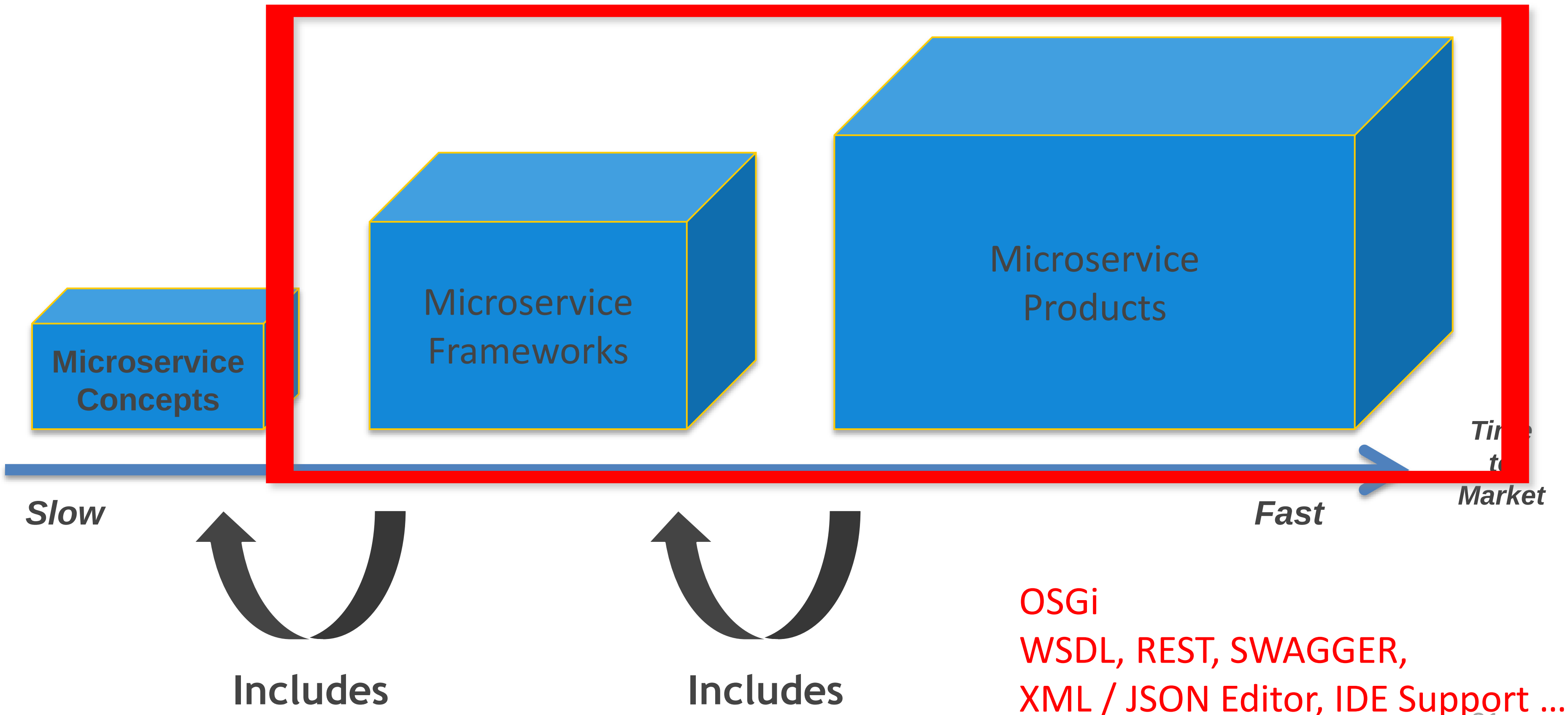
Other developers can easily access this information

Service contracts, and the ability for developers to discover them, serve that purpose.

TIBCO | Technologies for (Micro)Service Contracts

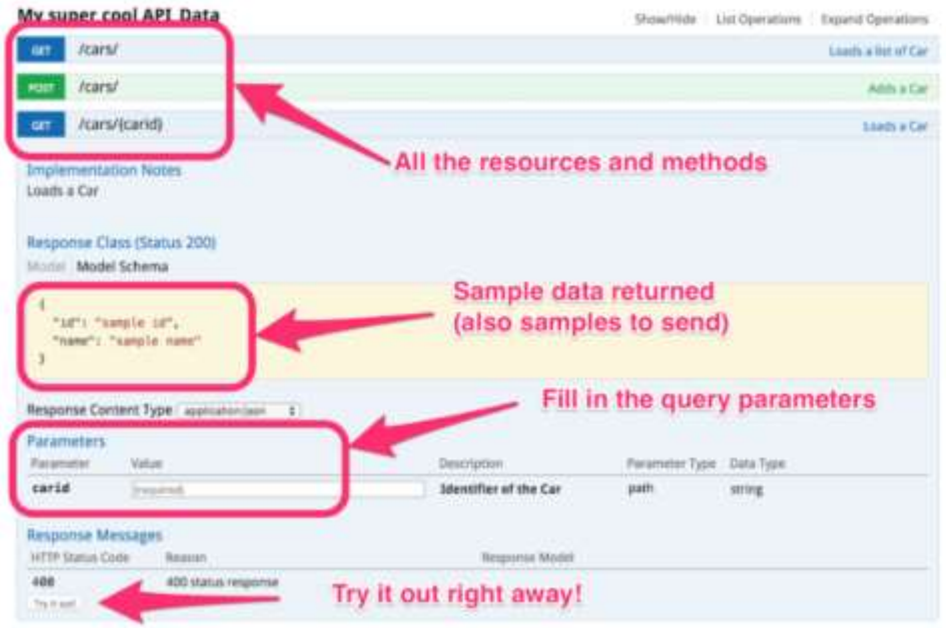
- Examples: IDL (CORBA), Java Interface, JMS Messages, SOAP, REST, …
- In Practice today:
  - SOAP: Internal, standards-based, XML Schema, easy mappings and transformations, performance no issue (anymore)
  - REST (i.e. RESTful HTTP without HATEOAS): External, XML or JSON, Good architecture for mobile devices (simplicity, separation of concerns, no state, uniform interface)
  - Messaging (e.g. WebSockets, MQTT): Good for millions of devices (IoT, sensors)
- De facto standard for Microservices as of today: REST
- Internet of Things will move Messaging forward!

Alternatives to Realize Microservices



OSGi
WSDL, REST, SWAGGER,
XML / JSON Editor, IDE Support …

Swagger – The de facto Standard for REST



My super cool API Data
Show/Hide | List Operations | Expand Operations
GET /cars/ Loads a list of Car
POST /cars/ Adds a Car
GET /cars/{carid} Loads a Car
All the resources and methods
Implementation Notes
Loads a Car
Response Class (Status 200)
Model Model Schema
Sample data returned (also samples to send)
Response Content Type application/json
Fill in the query parameters
Parameters
Parameter Value Description Parameter Type Data Type
carid (required) Identifier of the Car path string
Response Messages
HTTP Status Code Reason Response Model
400 400 status response
Try it out!
Try it out right away!

```
{
  "id": "sample id",
  "name": "sample name"
}
```

http://restlet.com/blog/2015/04/07/try-your-api-right-away-with-apisparks-built-in-swagger-ui/

"Swagger is a simple yet powerful representation of your RESTful API. With the largest ecosystem of API tooling on the planet, thousands of developers are supporting Swagger in almost every modern programming language and deployment environment. With a Swagger-enabled API, you get interactive documentation, client SDK generation and discoverability."

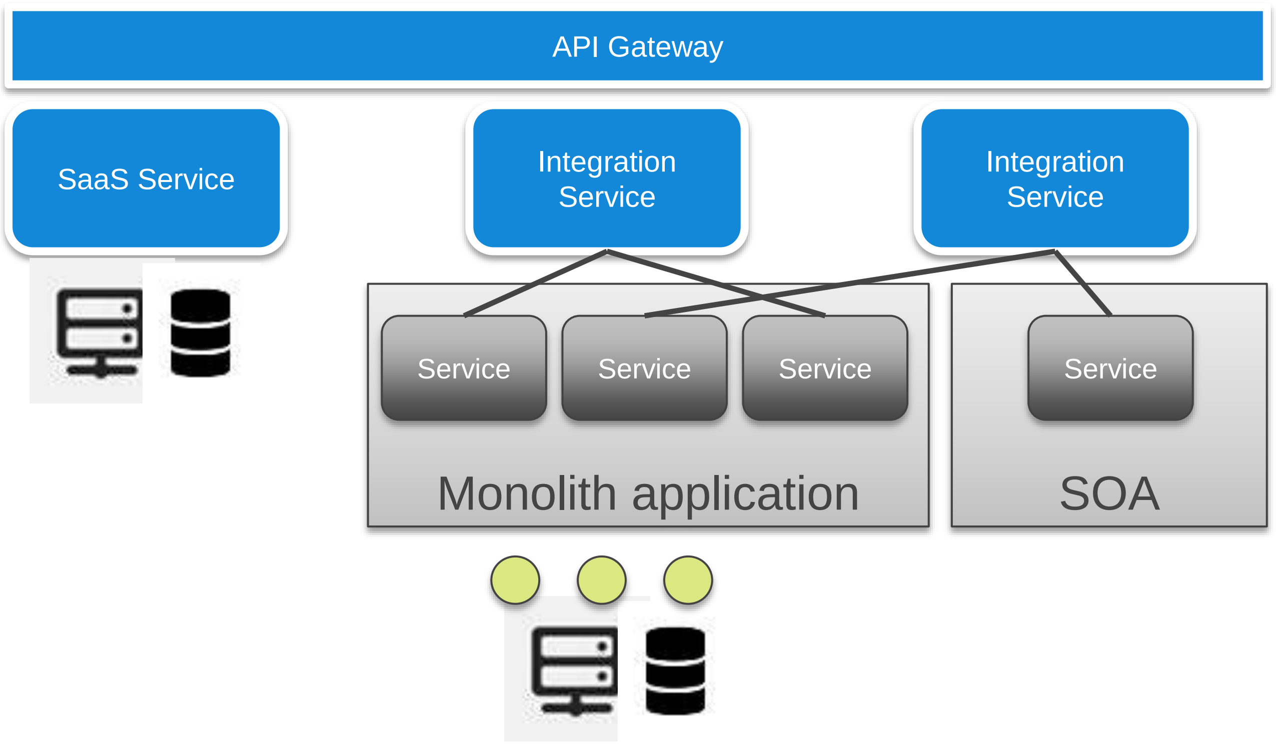
http://swagger.io/

TIBCO™ | Live Demo

Swagger for REST APIs in Action…

#2: Exposing new and existing Microservices

Services come in various forms



TIBCO™ |

NEW vs. EXISTING Services

NEW Services

- Build a service which uses (i.e. integrates) databases, files, applications, services, …

EXISTING Services

- Expose existing internal service via REST, SOAP, JMS …
- Use external services (SaaS)

Does not really matter… Integration is key!

TIBCO

Smart endpoints and dumb pipes

“When building communication structures between different processes, we've seen many products and approaches that stress putting significant smarts into the communication mechanism itself. A good example of this is the Enterprise Service Bus (ESB), where **ESB products often include sophisticated facilities for message routing, choreography, transformation, and applying business rules**.

The **Microservice community favours an alternative approach: smart endpoints and dumb pipes. Applications built from Microservices aim to be as decoupled and as cohesive as possible** - they own their own domain logic and act more as filters in the classical Unix sense - receiving a request, applying logic as appropriate and producing a response. These are choreographed using simple RESTish protocols rather than complex protocols such as WS-Choreography or BPEL or orchestration by a central tool.

The two protocols used most commonly are **HTTP request-response** with resource API's and **lightweight messaging**. The best expression of the first is

Be of the web, not behind the web

-- Ian Robinson”

http://martinfowler.com/articles/microservices.html#SmartEndpointsAndDumbPipes

TIBCO™

Smart endpoints and dumb pipes

“When building communication structures between different processes, we've seen many products and approaches that stress putting significant smarts into the communication mechanism itself. A good example of this is the Enterprise Service Bus (ESB), where **ESB products often include sophisticated facilities for message routing, choreography, transformation, and applying business rules**.

The **Microservice community favours an alternative approach: smart endpoints and dumb pipes. Applications built from Microservices aim to be as decoupled and as cohesive as possible** - they own their own domain logic and act more as filters in the classical Unix sense - receiving a request, applying logic as appropriate and producing a response. These are choreographed using simple RESTish protocols rather than complex protocols such as WS-Choreography or BPEL or orchestration by a central tool.

The two protocols used most commonly are **HTTP request-response** with resource API's and **lightweight messaging**. The best expression of the first is

Be of the web, not behind the web

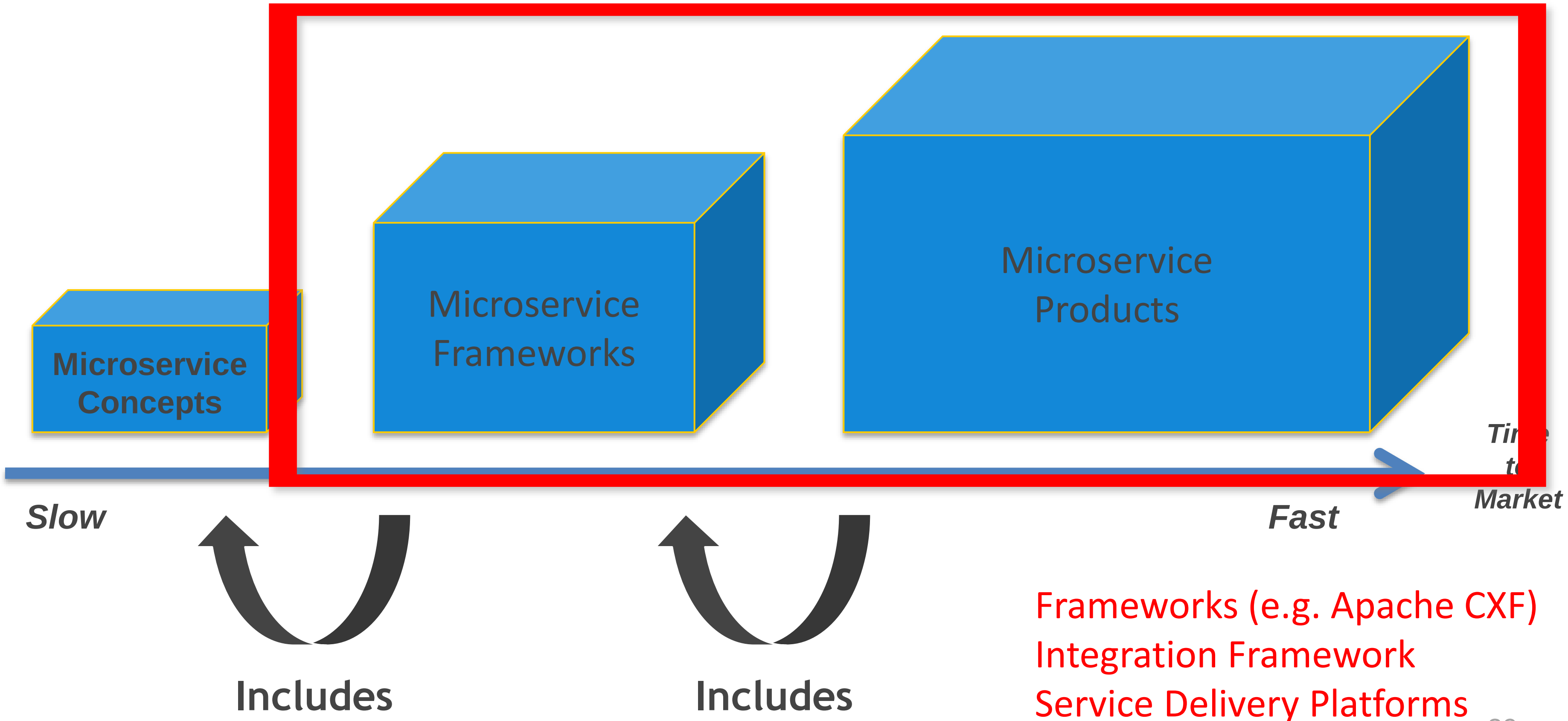
-- Ian Robinson”

Agreed!

However, be aware that you have to do “ESB tasks” (integration, routing, transformation, etc.) in the service then!

Why? It has to be done somewhere! Agree?

Alternatives to Realize Microservices



Frameworks (e.g. Apache CXF)
Integration Framework
Service Delivery Platforms

TIBCO™ | Build new services with frameworks and tools

Service Frameworks

- Java EE => JAX-RS /-WS (Apache CXF)
- .NET => WCF
- Python
- Ruby
- “you-name-it”

Integration Framework

- Apache Camel (JVM)
- Spring Integration (JVM)
- NServiceBus (.NET)

Service Delivery Platform (formerly often called ESB)

- TIBCO BusinessWorks
- Talend ESB
- WSO2 ESB
- “you-name-it”

Integration Framework - Apache Camel

```
from("direct:processOrder")
    .to(”javaBean:orderService?method=process”)
    .to(“jdbc:myDatabase:myTable?sql=update…”)
    .to(”jms:queue:order.out");
```

1) → Implement Microservices logic in so called “routes” …

2) ↓

3) ↘

… and expose it as service, e.g. REST, SOAP or JMS.

```
from(”rest:serverDummyURL")
  .to("log:TEST?showAll=true”)
  .direct(processOrder);
```

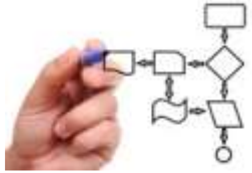
```
from(”jms:myQueue:in")
  .to("log:TEST?showAll=true");
  .direct(processOrder);
```

TIBCO | Live Demo

Apache Camel

Apache Camel in Action...

TIBCO | Service Delivery Platform – TIBCO BusinessWorks



Accelerate Time to Results

Zero-code Integration

Non-stop Dev-Deploy

Visual Debugger

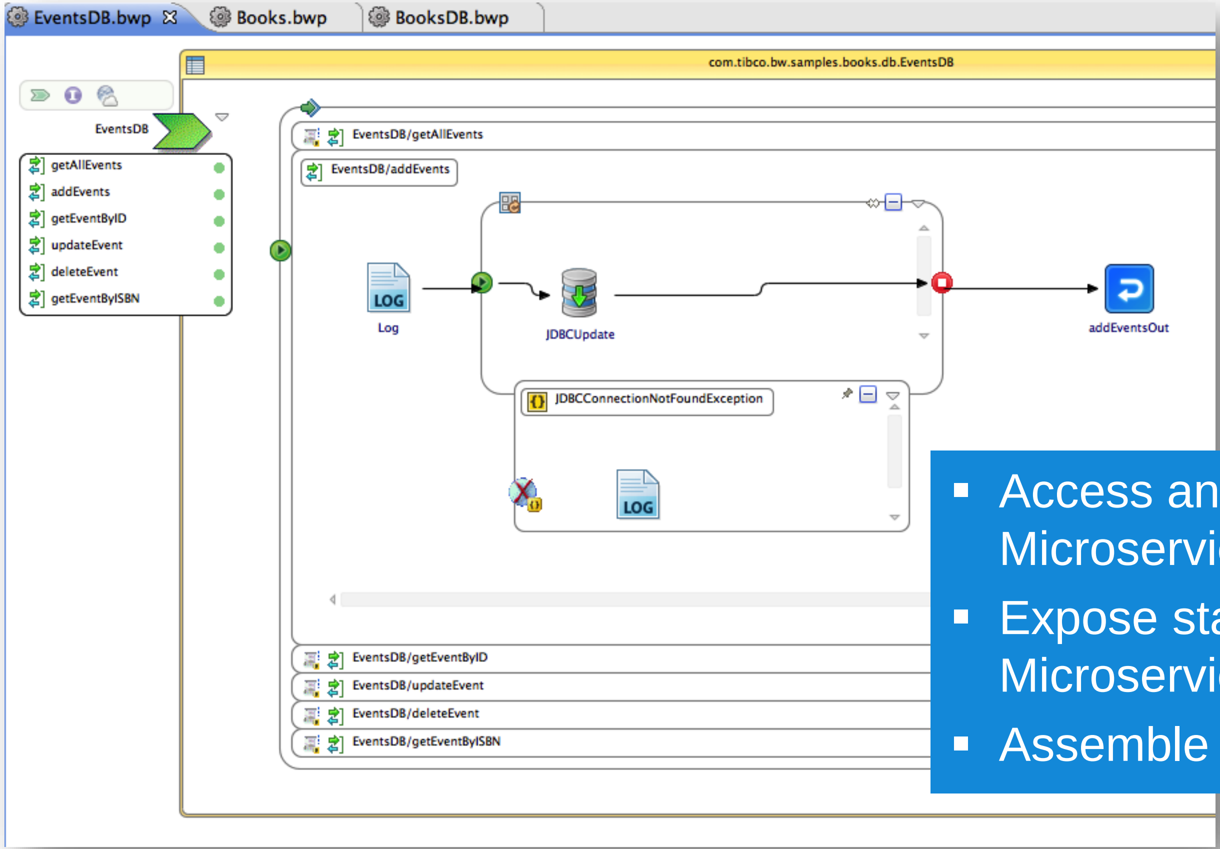
Simple Sophisticated Modeling

Multi-op Processes

Conversations

Event Handlers

Integration as foundation of Microservices

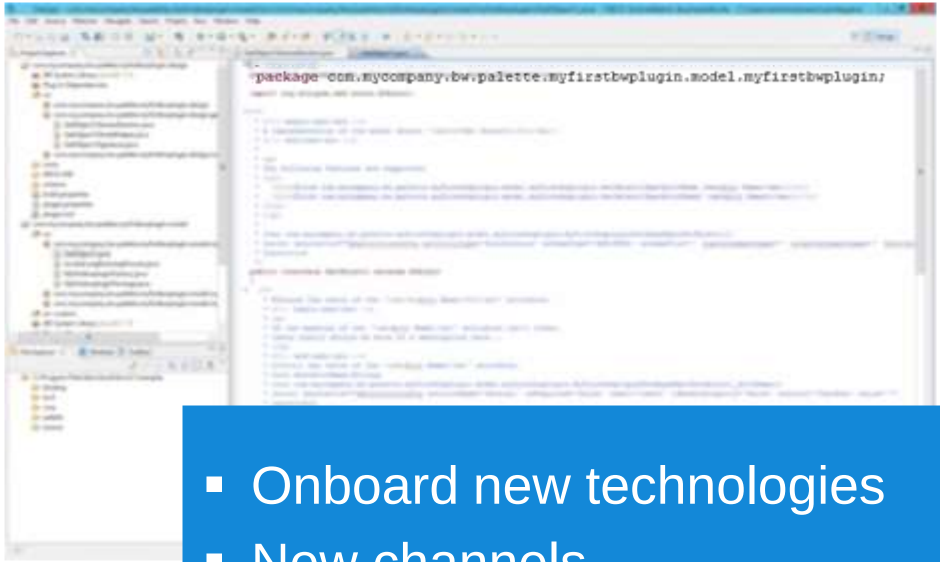


- Access any data to use in Microservices
- Expose standard transport from Microservices
- Assemble new Microservices

Leverage any technology to create Microservices

Abstract complex APIs using:

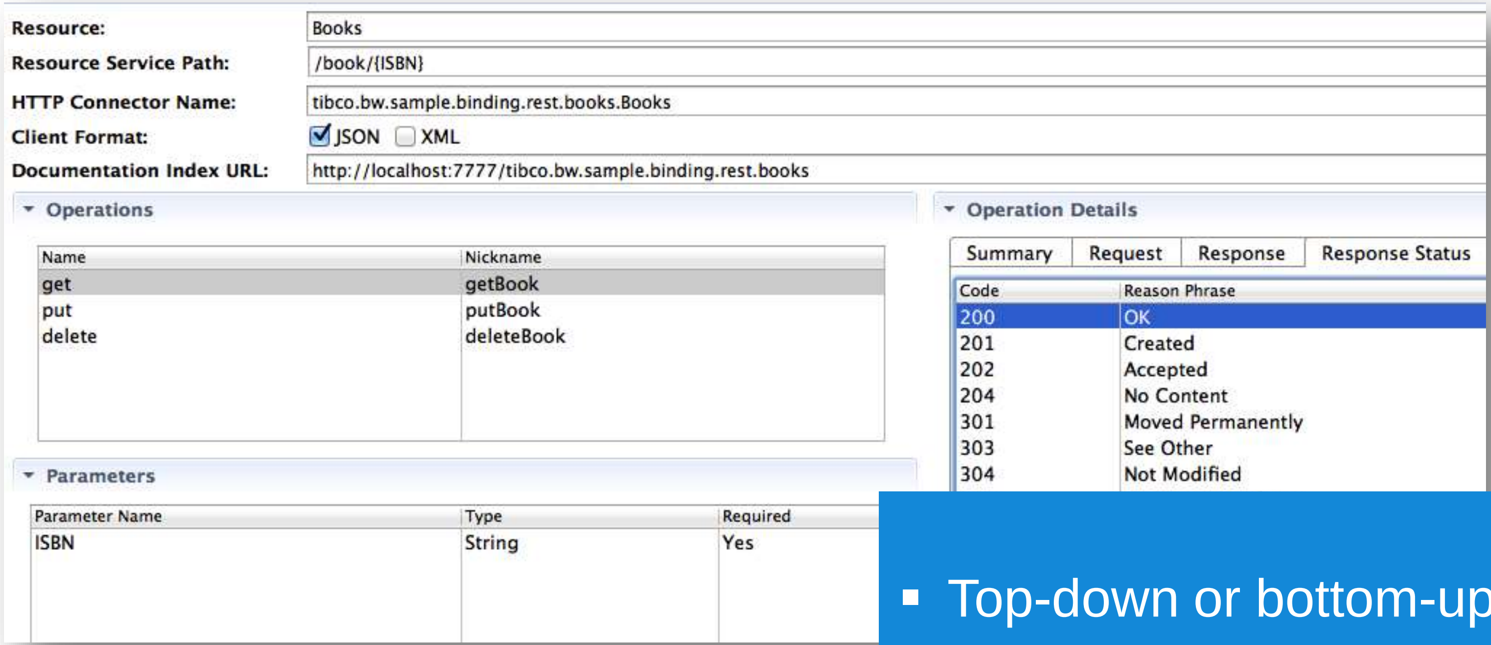
- Standard connectors
  - File, JDBC, SOAP, REST, JMS, etc.
- Application connectors
  - SaaS (SFDC, Marketo), SAP, Big Data, Mobile, legacy applications, etc.
- Plugin development kit
- Programming languages
  - Java, Scala, Ruby, etc.



```
package com.mycompany.bw.palette.myfirstbwplugin.model.myfirstbwplugin;
```

- Onboard new technologies
- New channels
- New data sources

Create REST APIs service In Under A Minute



Resource:	Books
Resource Service Path:	/book/{ISBN}
HTTP Connector Name:	tibco.bw.sample.binding.rest.books.Books
Client Format:	[✓ JSON] XML
Documentation Index URL:	http://localhost:7777/tibco.bw.sample.binding.rest.books

Operations

Name	Nickname
get	getBook
put	putBook
delete	deleteBook

Parameters

Parameter Name	Type	Required
ISBN	String	Yes

Operation Details

Code	Reason Phrase
200	OK
201	Created
202	Accepted
204	No Content
301	Moved Permanently
303	See Other
304	Not Modified

- Top-down or bottom-up modeling
- Automatic docs and testing web UI

TIBCO | Live Demo

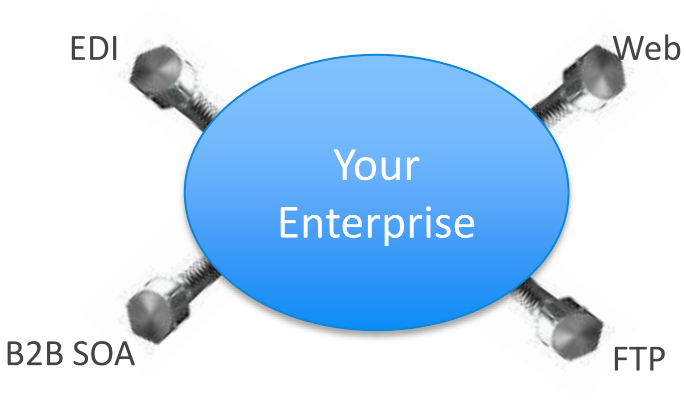
TIBCO BusinessWorks

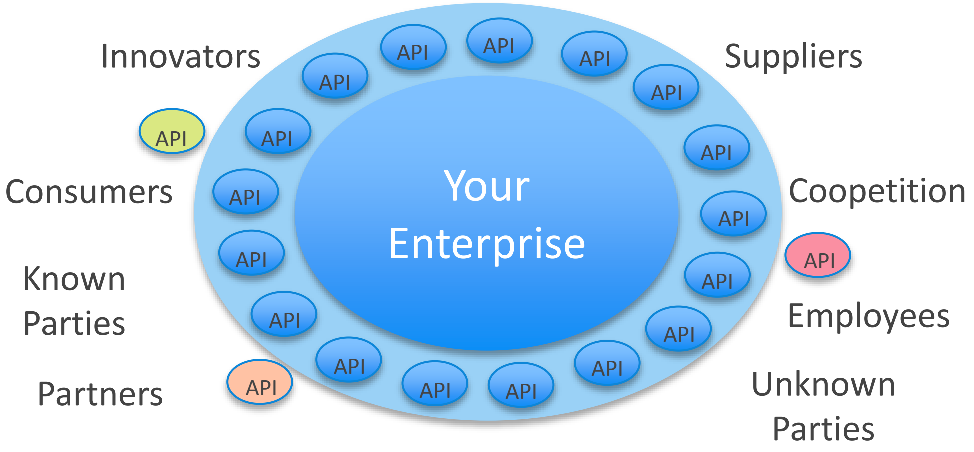
TIBCO BusinessWorks 6 in Action...

#3: Discovery of Services

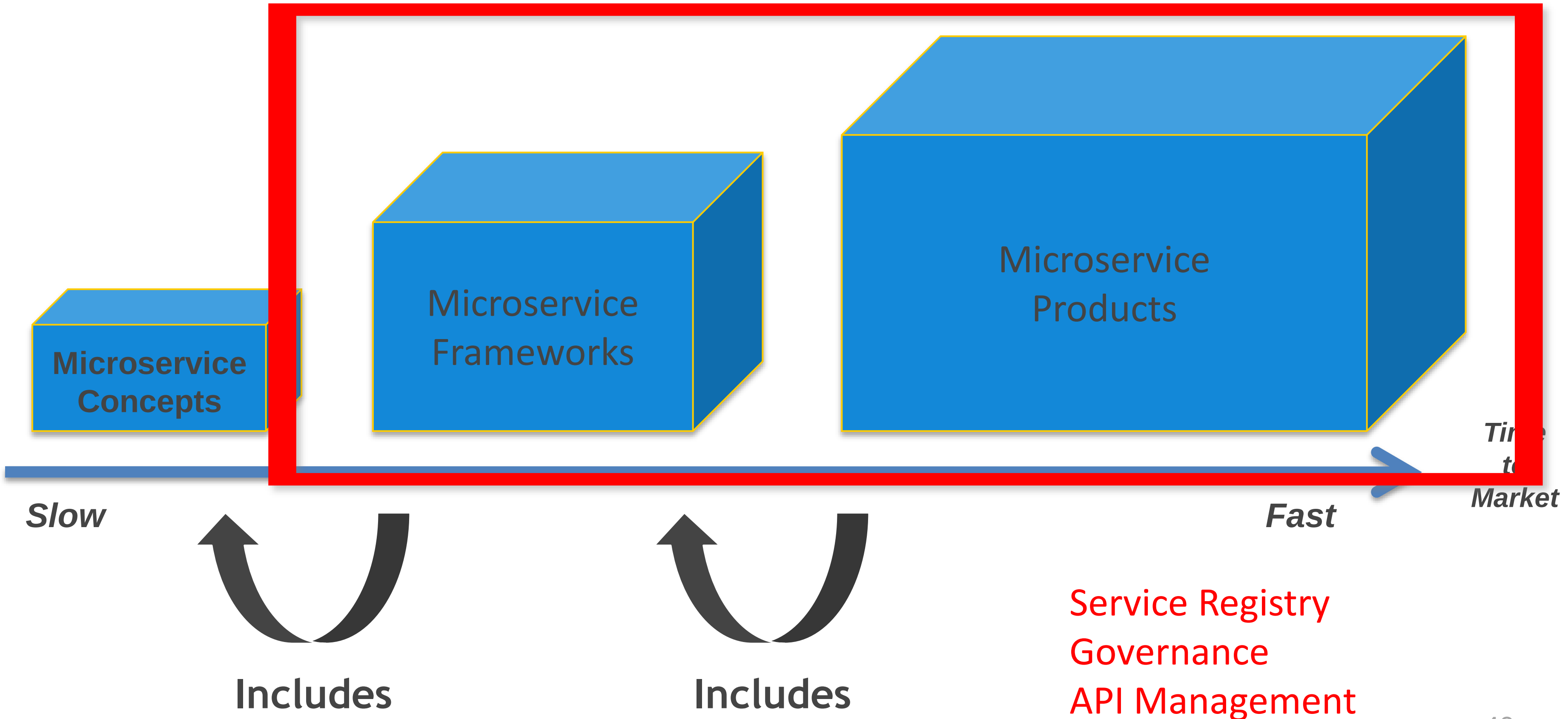
The new "Open API" Economy

Closed APIs	Open APIs
Pre-defined integration points	Opportunistic access points
Limited, trusted partners	Many partners, untrusted
Strictly constrained interactions	Encourage new ideas
Enterprise friendly	Developer friendly





Alternatives to Realize Microservices



Service Registry
Governance
API Management

API Management - JBoss APIMAN

Open Source API Management

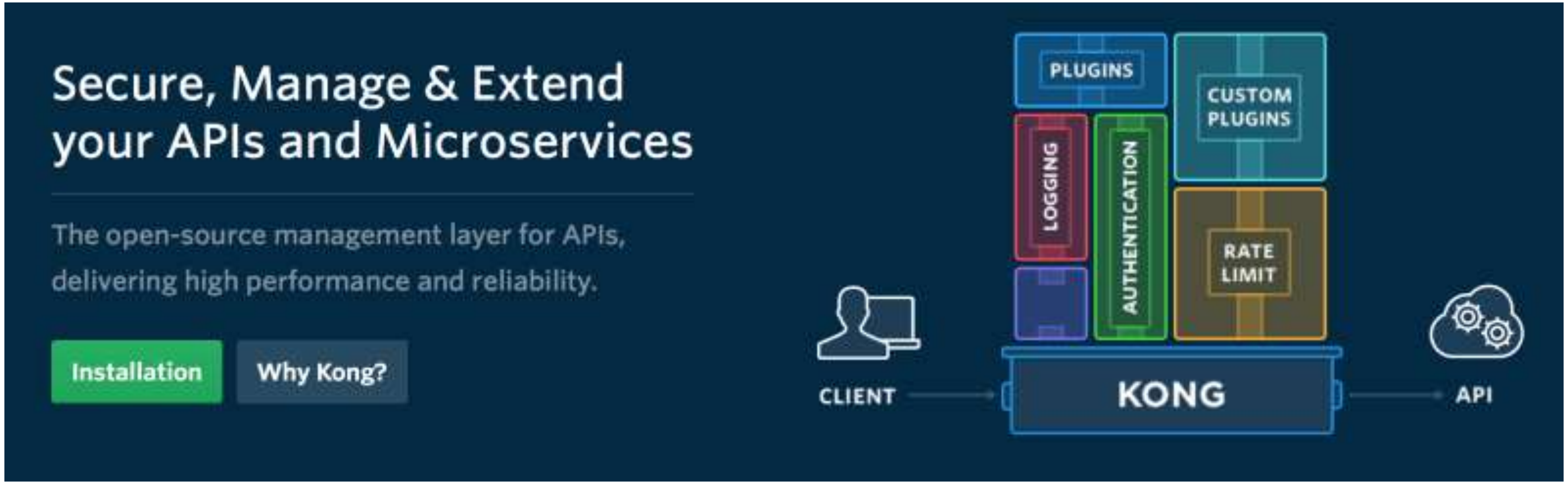
The apiman project brings an open source development methodology to API Management, coupling a rich API design & configuration layer with a blazingly fast runtime.

(Version 1.1.2.Final)

Clone or Fork apiman on GitHub

http://www.apiman.io

TIBCO | API Management - KONG


Secure, Manage & Extend your APIs and Microservices

The open-source management layer for APIs, delivering high performance and reliability.

http://getkong.org/

TIBCO™ |

API Management – WSO2 API Manager

WSO2 API Manager

- A unique storefront experience to help easily find, evaluate, and test APIs online
- Embedded analytics to measure APIs' success
- Built-in lifecycle and version management
- Supports the OAuth security standard and gives you control over who can access your APIs

http://wso2.com/landing/ppc/api-manager

API Management – TIBCO API Exchange

Cloud Based Or On-Premise

API Portal

- Developer self-service
- API Lifecycle
- API Monetization

API Gateway

- Security & Access Control
- Event Based Policy Mgt.
- Federated Internet Scale

API Analytics

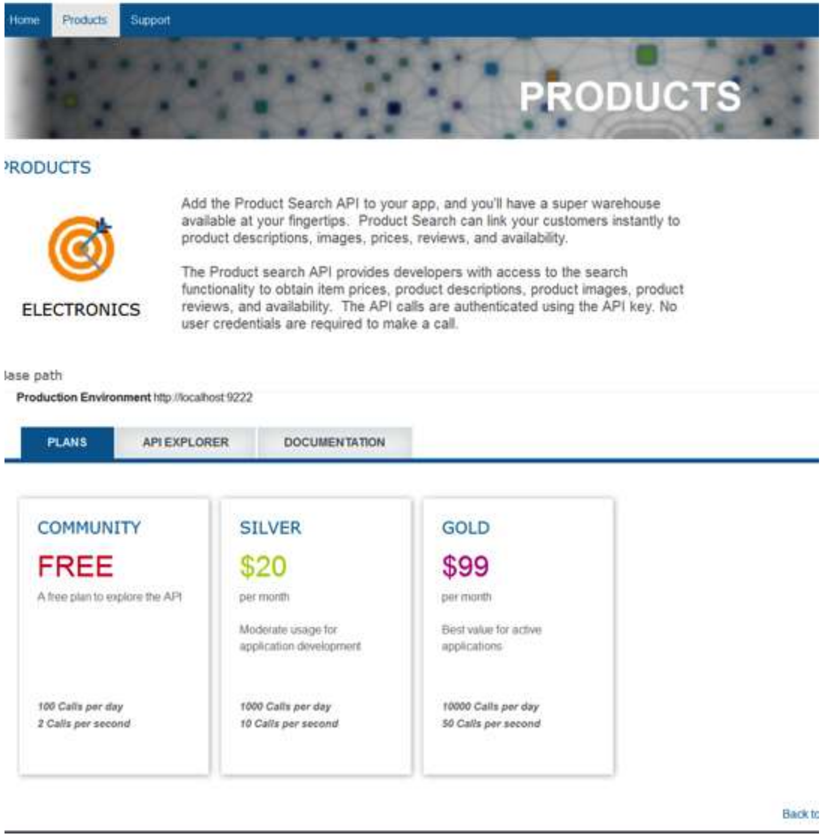
- Reporting / Visualization
- SLA’s & KPI’s
- Full Auditing

TIBCO™

TIBCO API Exchange - Portal

The API Portal allows all developer to discover Microservices and their contracts, read documentation, and test the APIs.

- Discovery of Services
- Service Catalog
  - SOAP and REST services
  - Interactive and unstructured docs
  - Authentication by API Key, OAuth
  - Create REST proxy from portal
- Contracts
  - Pre-defined or custom QoS
  - Rate/day, rate/second
  - Approval workflow (or automatic)
  - Plans can route to different targets



RODUCTS

ELECTRONICS

Add the Product Search API to your app, and you'll have a super warehouse available at your fingertips. Product Search can link your customers instantly to product descriptions, images, prices, reviews, and availability.

The Product search API provides developers with access to the search functionality to obtain item prices, product descriptions, product images, product reviews, and availability. The API calls are authenticated using the API key. No user credentials are required to make a call.

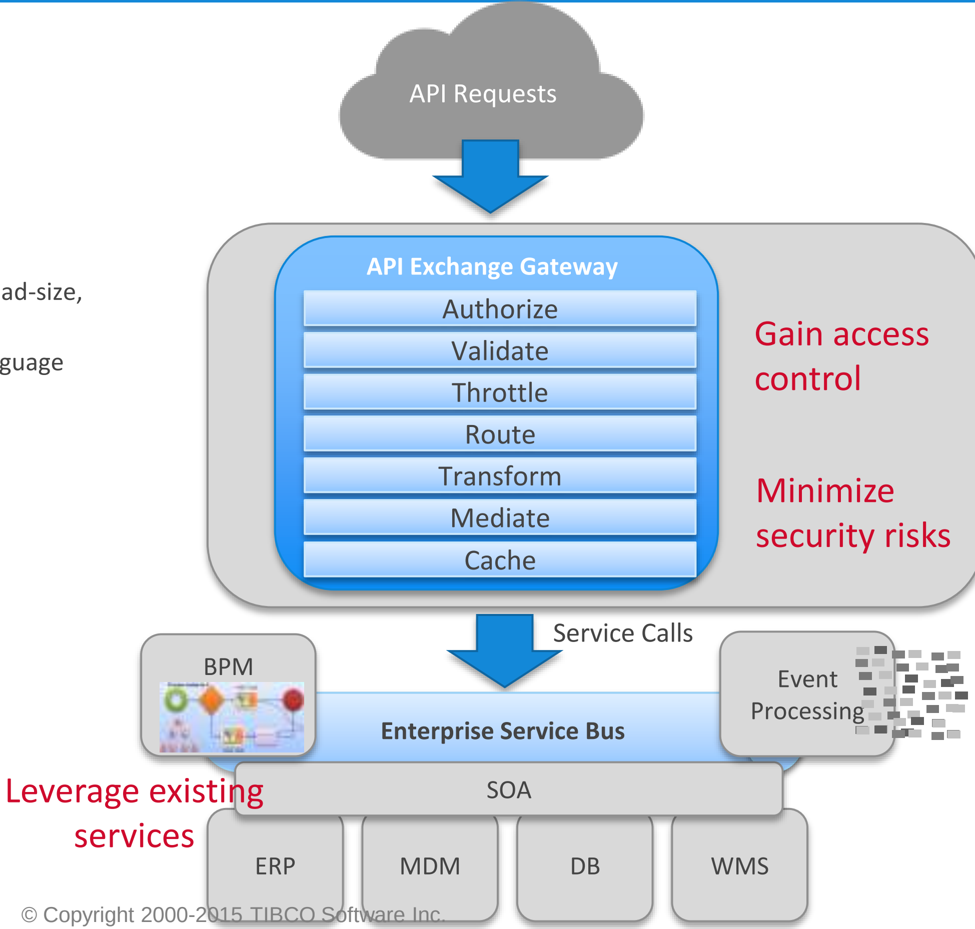
ase path

Production Environment http://localhost:9222

COMMUNITY	SILVER	GOLD
FREE	$20	$99
A free plan to explore the API	per month	per month
	Moderate usage for application development	Best value for active applications
100 Calls per day	*1000 Calls per day*	*10000 Calls per day*
2 Calls per second	*10 Calls per second*	*50 Calls per second*

TIBCO API Exchange - Gateway

- **Authorization - *whose* requests**
  - Access control granularity down to service endpoint
  - Single-edit configuration changes through web user interface
  - Security standards: LDAP, SAML, OAuth, WSPolicy, etc.
- **Throttling - *when* requests are handled**
  - Rate & High-Water Mark, Quota, Time-of-Day, Error-rate/Payload-size, Group Logical, Traffic shaping
  - Policies and throttles can be extended with declarative rule language in Studio
- **Routing - *where* requests are handled**
  - Single-edit configuration through web user interface
  - In-line transformation through configuration
  - Orchestration logic can be hot-deployed
  - By operation, version, size, time of day, etc.
- **Mediation - *how* requests are handled**
  - 'Flow' logic
  - Transformation and Validation logic
  - Caching logic



TIBCO™ | TIBCO API Exchange - Analytics

Understand usage and performance through interactive reporting for both API providers and consumers

API Consumer

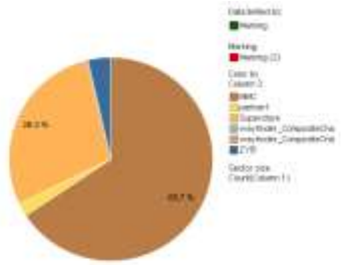
- Application Performance
- Debugging
- Usage/Limit Monitoring

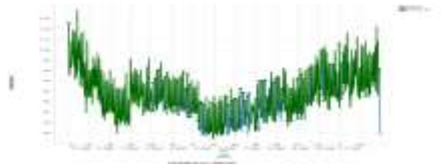
Measure and improve application performance

API Provider

- API Performance
- Auditing
- Operational Monitoring

Measure and improve on the success of API initiatives





TIBCO | Live Demo

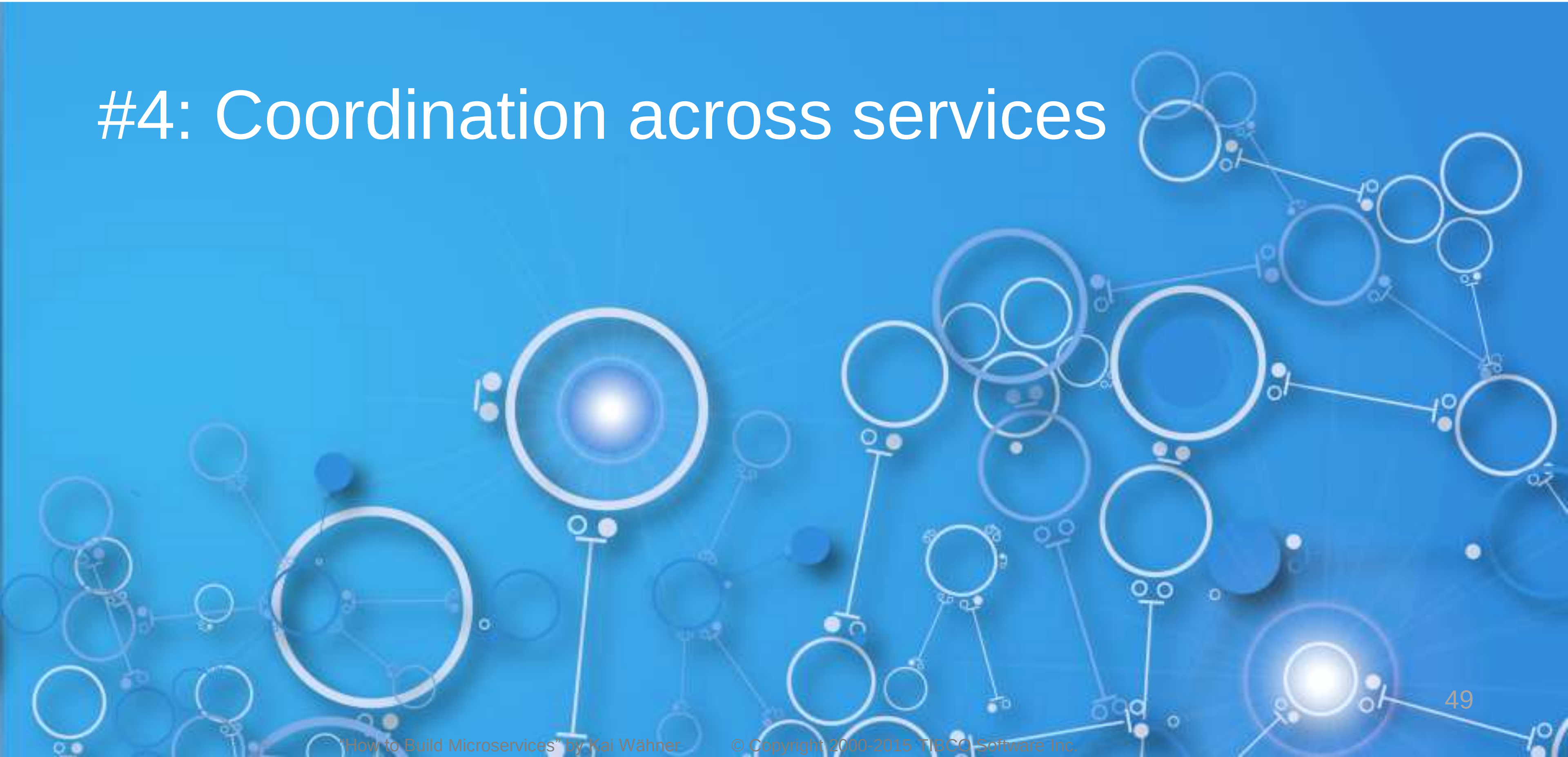
TIBCO

API Exchange

TIBCO API Exchange in Action...

TIBCO™ | Requirements for Microservices Architecture

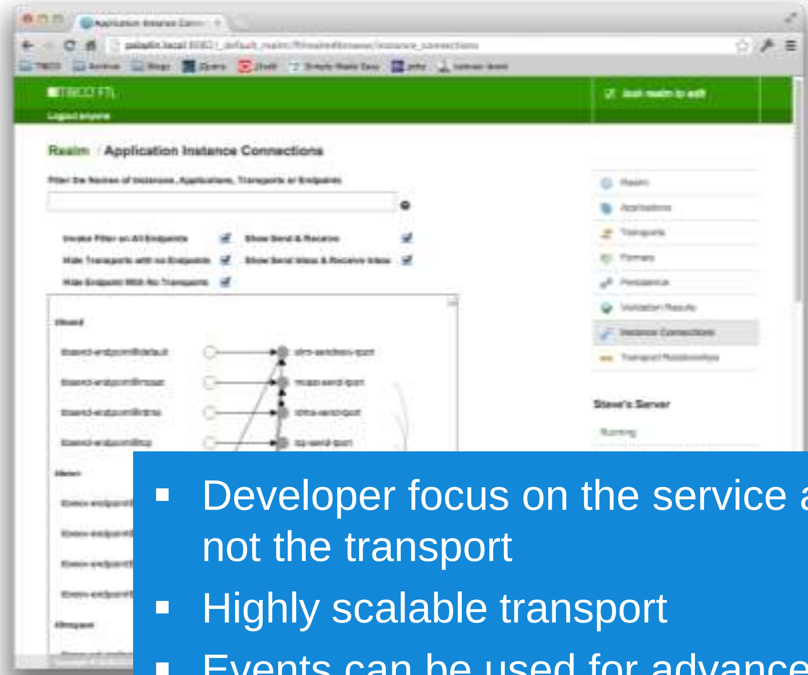
#4: Coordination across services



TIBCO™

Smart endpoint, dumb pipes… → Messaging

- API is not just exposing a service via SOAP / REST!
- JMS, MQTT, AMQP, WebSockets, CoAP, DSS, …
- TIBCO FTL
  - Peer-to-Peer messaging with the power, flexibility and control of a server-based store and forward design
  - Frees up application developers to focus on message processing by decoupling message distribution from application development

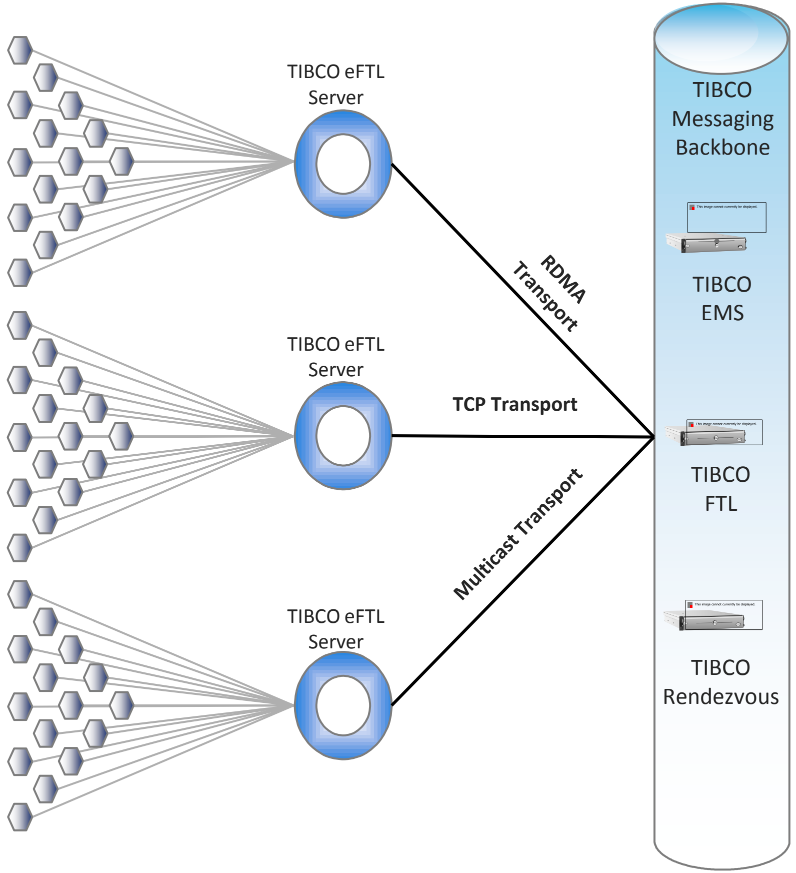


- Developer focus on the service and not the transport
- Highly scalable transport
- Events can be used for advanced logic

Extending Services to Mobile Applications

- eFTL servers provide increased scalability for web and mobile based applications
- Mobile clients communicate directly with TIBCO eFTL servers over HTML5 Web-sockets. Native Mobile Application Support
  - Objective-C (iOS)
  - Android
  - JavaScript

- Communication between apps and devices,
- Communication from apps and devices to your Microservices
- Robustness at the scale of the Internet of Everything



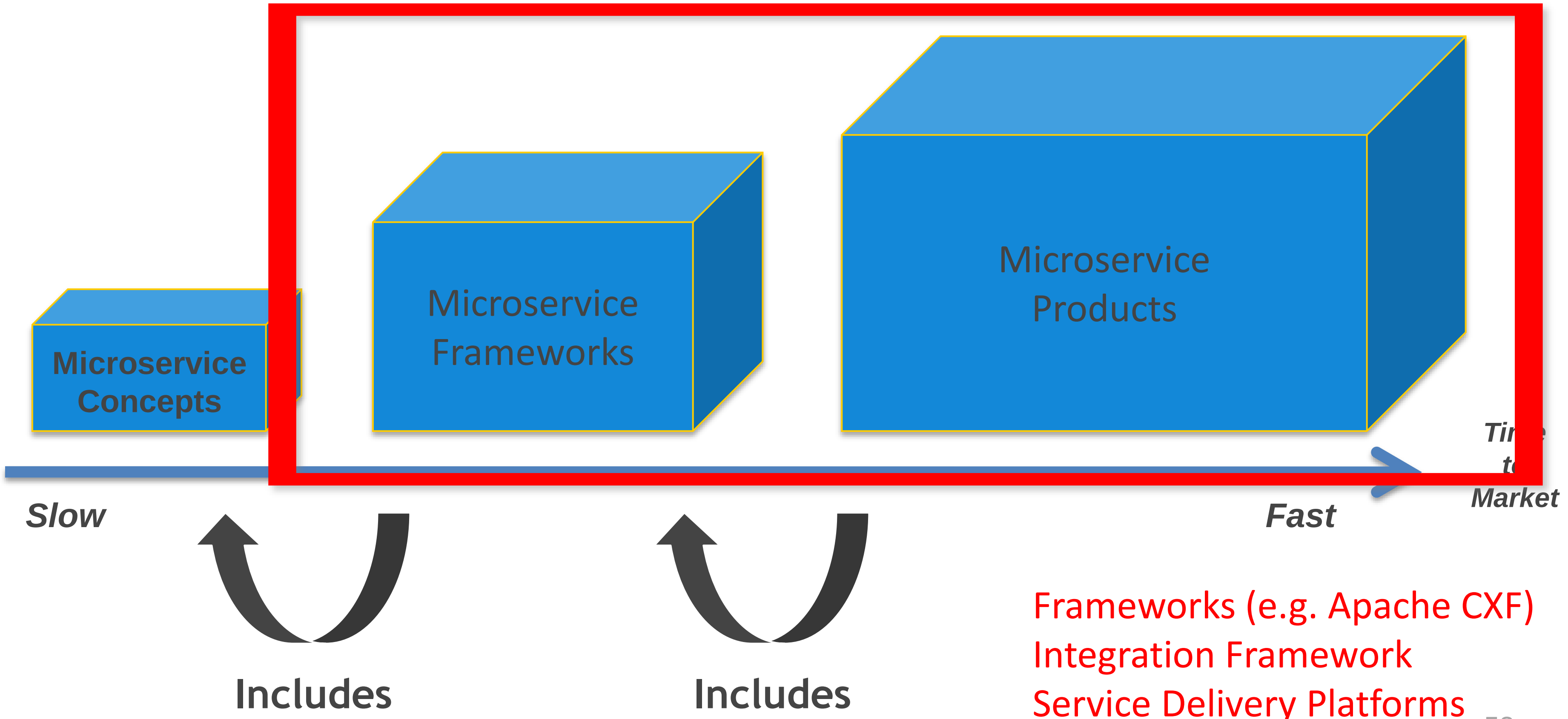
TIBCO™

Coordination across services via NEW services ?!

Smart service, dumb pipe (no ESB in the middle)…

How to coordinate?

Alternatives to Realize Microservices



Frameworks (e.g. Apache CXF)
Integration Framework
Service Delivery Platforms

TIBCO™

Apache Camel for Coordination across Services

```
from(”rest:serverDummyURL")
  .to("log:TEST?showAll=true”)
  .direct(processOrder);
```

```
from(”jms:myQueue:in")
  .to("log:TEST?showAll=true");
  .direct(processOrder);
```

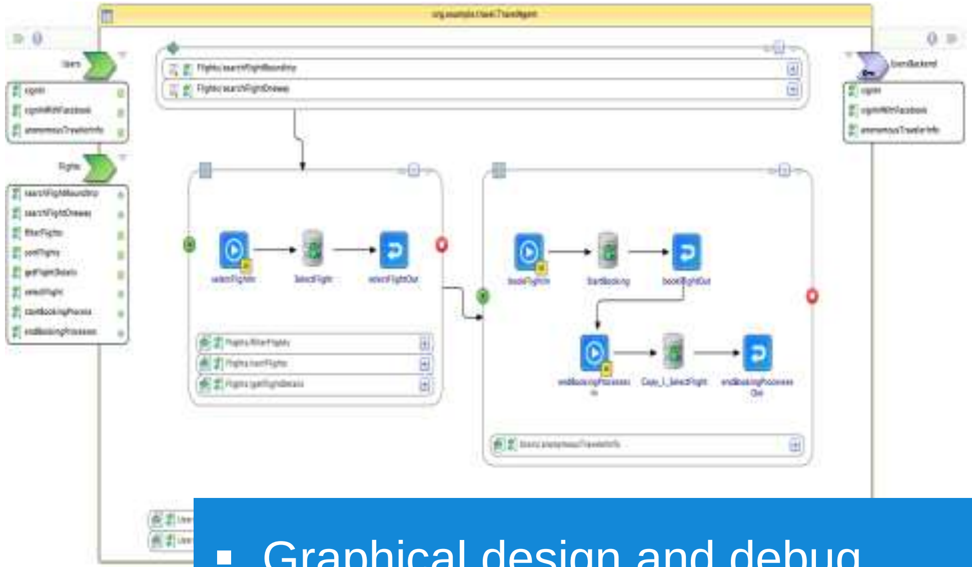
1) → Existing Microservices

2) ↓ Coordination with another route

```
from(”rest:serverRequestURL")
        .choice()
          .when(header("foo").isEqualTo("bar"))
            .to(”rest:serverDummyURL")
          .when(header("foo").isEqualTo("cheese"))
            .to(”jms:myQueue:in")
          .otherwise()
            .to(”log:errorHandler");
```

TIBCO | BusinessWorks for Coordination across Services

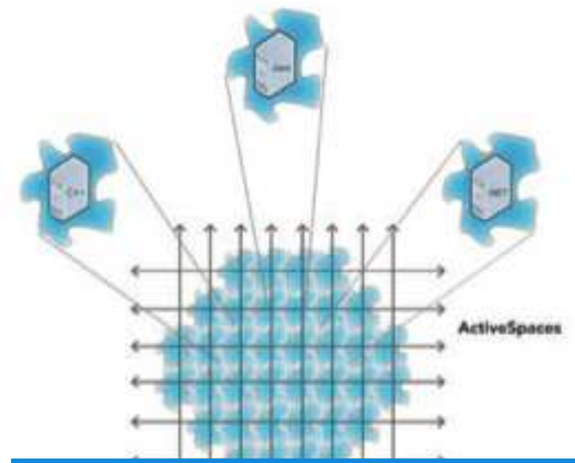
- Apps/business services are composed from Microservices
- Some Microservices can be composed to accelerate developments



- Graphical design and debug
- Stateful or stateless
- Service or event driven

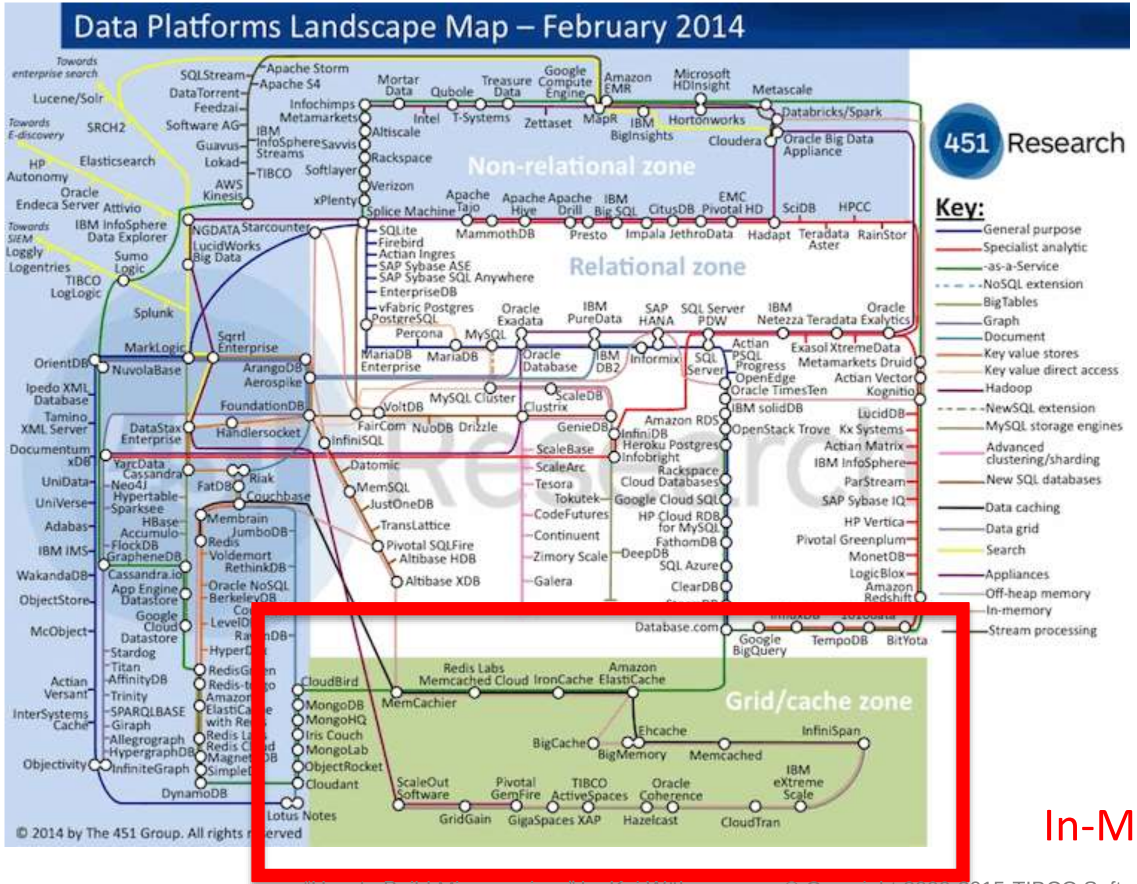
In-Memory Data Grid

- Share data and store context in an distributed in-memory data grid
  - Provides data storage for services
- Speed up communication and coordination between Microservices



- Provide a common repository for services managing the same business objects
- Share change of context / state as events

In-Memory Data Grid – Market Overview

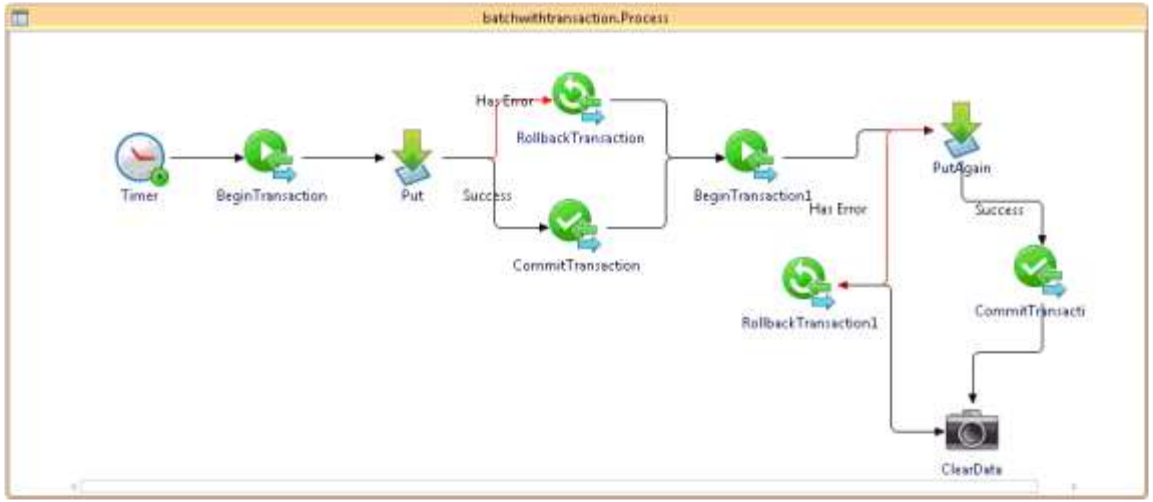


http://blogs.the451group.com/information_management/2015/03/18/updated-data-platforms-landscape-map-february-2015/

In-Memory Data Grids

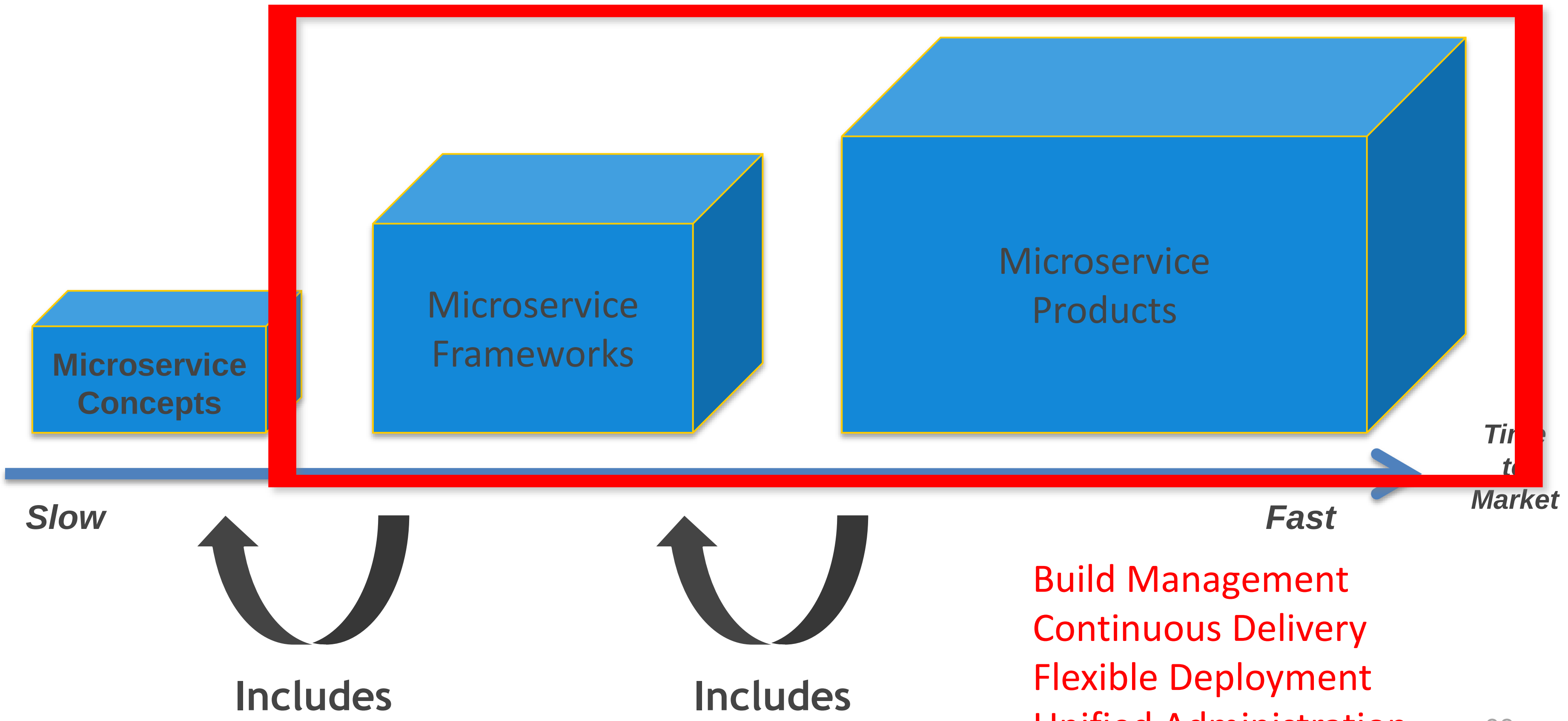
Integration with TIBCO BusinessWorks

TIBCO ActiveMatrix BusinessWorks Plug-in for ActiveSpaces
to utilize all the benefits of TIBCO ActiveSpaces without any coding



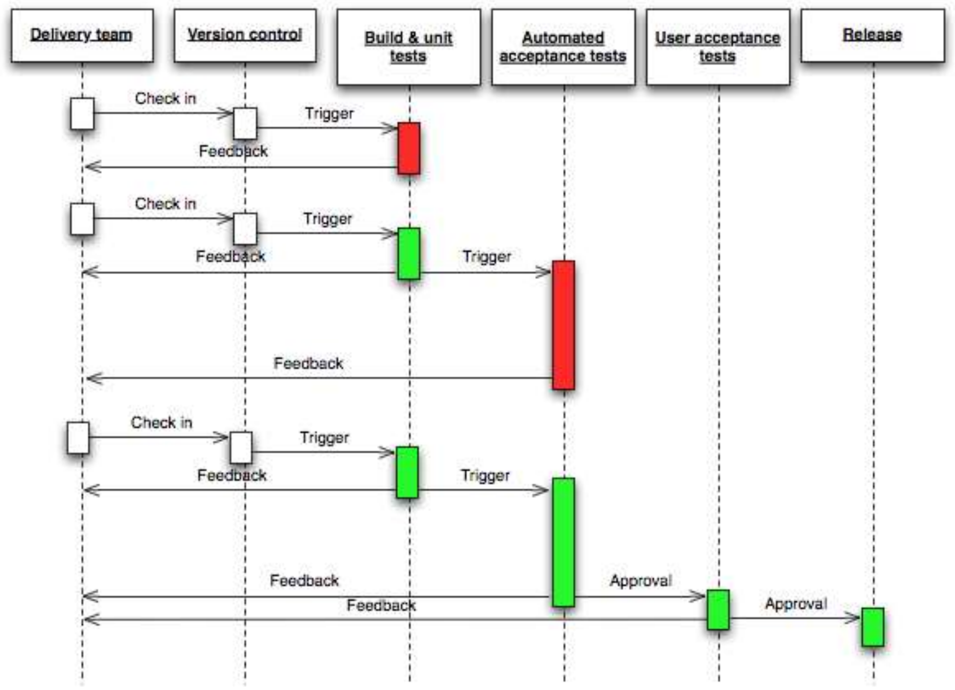
#5: Managing complex deployments and their scalability

Alternatives to Realize Microservices



Build Management
Continuous Delivery
Flexible Deployment
Unified Administration

TIBCO | Continuous Delivery



http://en.wikipedia.org/wiki/Continuous_delivery

Benefits

- Accelerated Time to Market
- Building the Right Product
- Improved Productivity and Efficiency
- Reliable Releases
- Improved Product Quality
- Improved Customer Satisfaction

Combined with “Cloud”

- Private / Public / Hybrid PaaS
- Flexible Infrastructure
- Elasticity

TIBCO

Frameworks and Products for Continuous Delivery and DevOps

- Build Management
  - Ant, Maven, Gradle
- Continuous Integration
  - Jenkings, Bamboo
- Continuous Delivery
  - Chef, Puppet, Salt
- Deployment (Elastic VMs / Cloud / Containers)
  - Amazon Web Services, Microsoft Azure, CloudFoundry
  - VMware, Openstack, Vagrant
  - Docker, Spring Boot

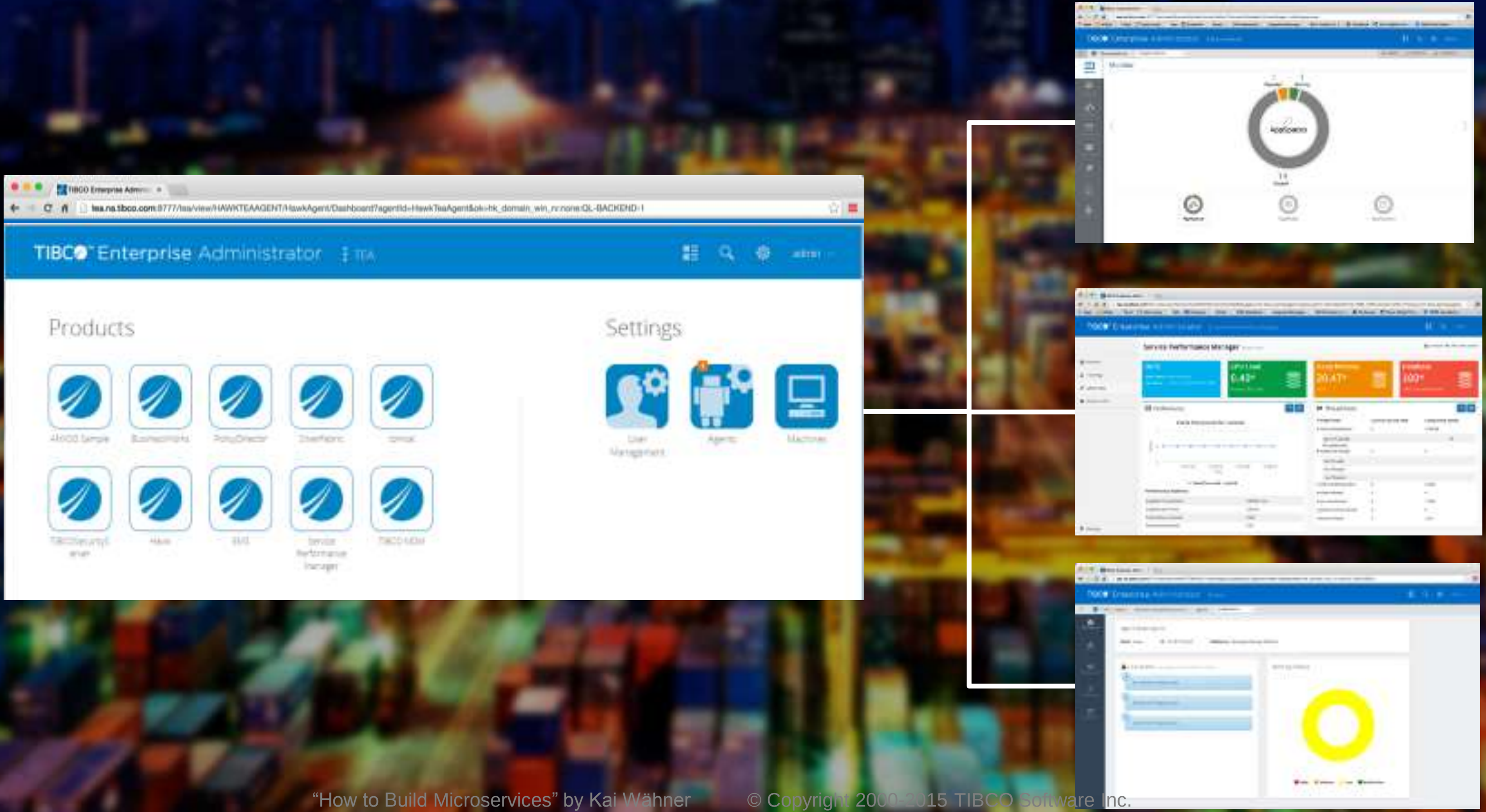
TIBCO™ | TIBCO Silver Fabric

- **DevOps in the TIBCO Universe**
  - Out-of-the-box support for TIBCO products such as BusinessWorks
  - Complementary (not XOR!) to build, continuous integration and delivery, cloud, container and VM tools (see last slide)!
- **Continuously** deploy, configure and manage your **applications and middleware**, on premise or in the cloud.
- DevOps – Continuous Integration / Delivery
  - Configuration Manager for Global Variables
  - End-to-End Scripting, Automation & Visibility
- Manages quality of deployed applications
  - Ports Management & Elastic Load Balancer
  - Dashboard & Full Visibility
  - SLA based auto scaling & elasticity

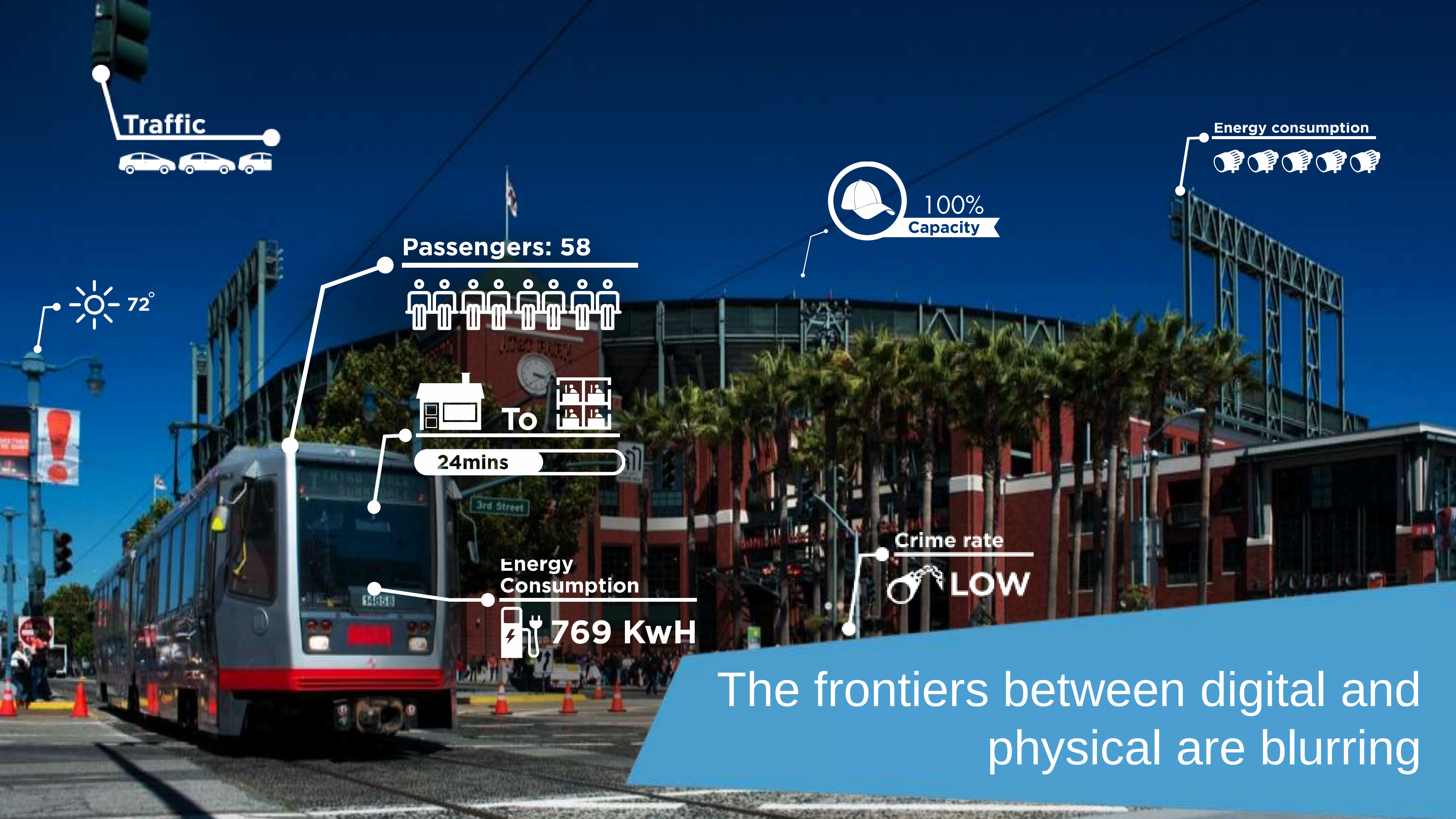
Self Service

Administration

TIBCO TEA: Unified Administration

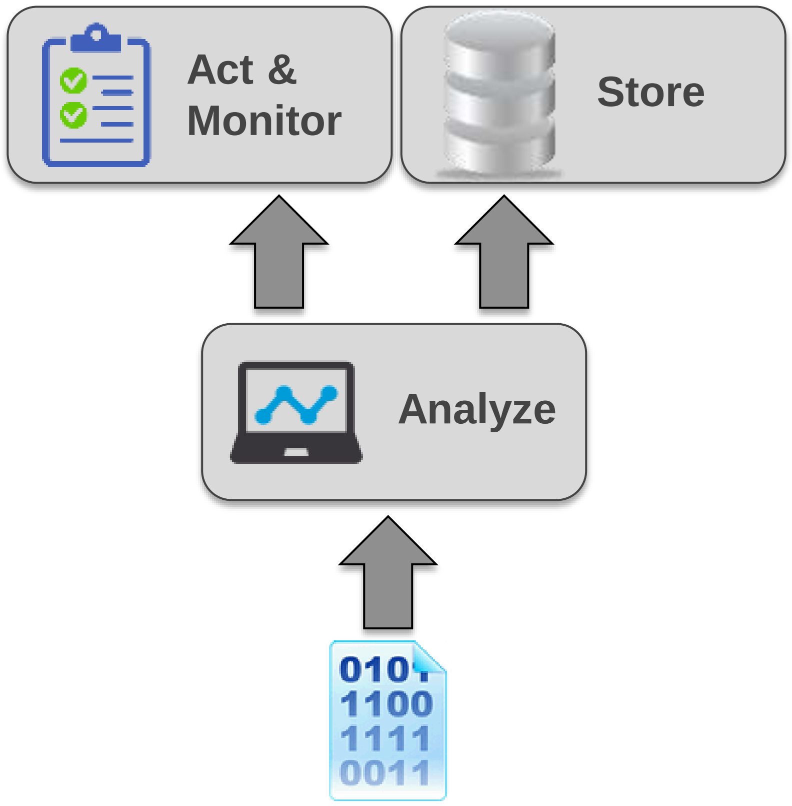

TIBCO Enterprise Admin
tea.na.tibco.com:8777/tea/view/HAWKTEAAGENT/HawkAgent/Dashboard?agentId=HawkTeaAgent&ok=hk_domain_win_nr:none:QL-BACKEND-1
TIBCO Enterprise Administrator
TEA
admin
Products
AKIOO Sample
BusinessWorks
PolicyDirector
SilverFabric
Spotfire
TIBCOSecurityServer
Hawk
EMS
Service Performance manager
TIBCO MDM
Settings
User Management
Agents
Machines
Service Performance Manager

#6: Visibility and Correlation across Services



The frontiers between digital and physical are blurring

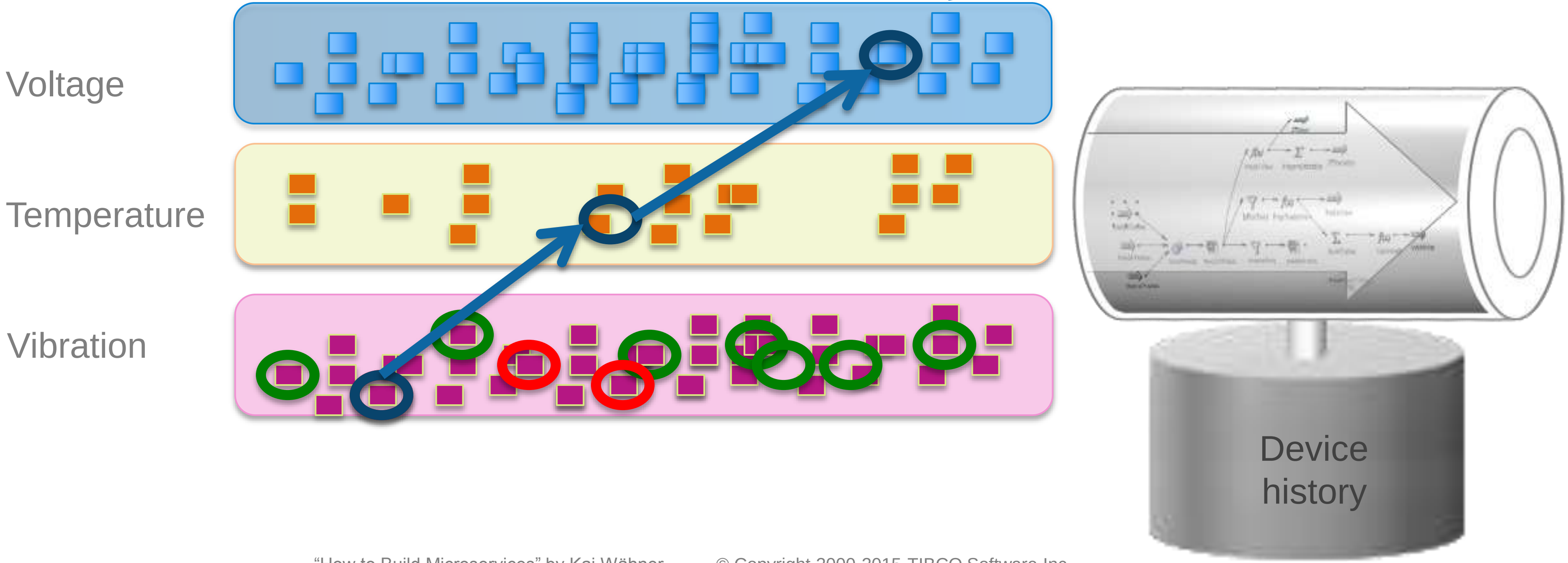
The New Era: Fast Data Processing



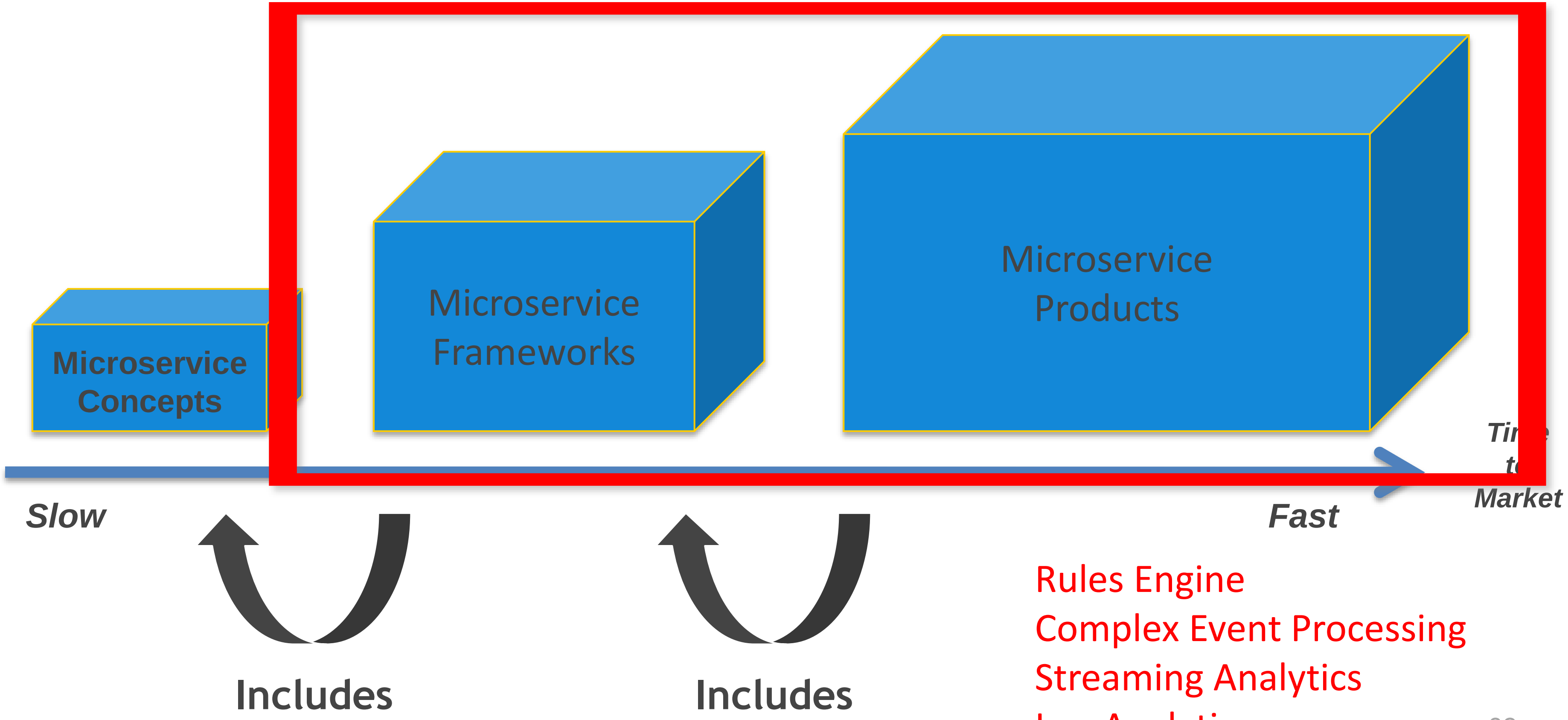
- Events are **analyzed and processed in real-time** as they arrive.
- Decisions are **timely, contextual, and based on fresh data**.
- **Decision latency is eliminated**, resulting in:
  - ✓ Superior Customer Experience
  - ✓ Operational Excellence
  - ✓ Instant Awareness and Timely Decisions

Event Processing (Correlation of Events)

<u>**Temporal analytic**</u>: "If **vibration spike** is followed by **temp spike** then **voltage spike** [within 12 minutes] then flag **high severity alert**."

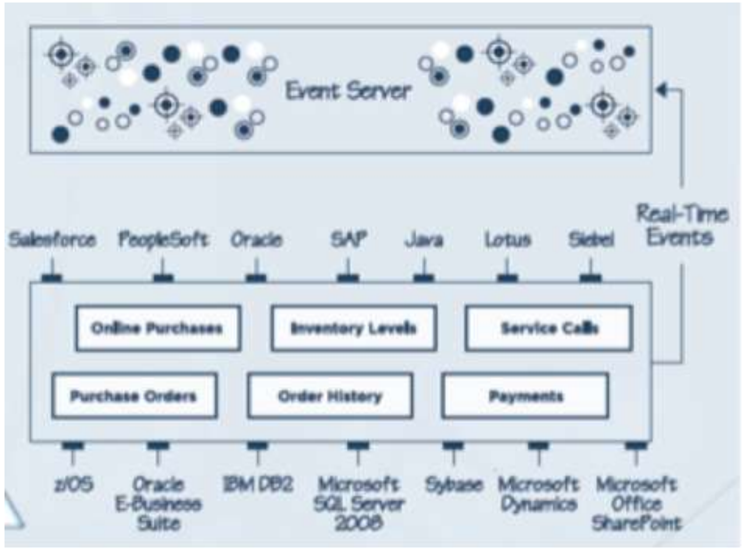


Alternatives to Realize Microservices



Rules Engine
Complex Event Processing
Streaming Analytics
Log Analytics

The need for a “Bus”?

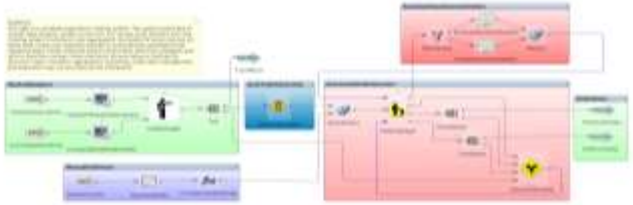


Event correlation is the requirement, where you really need a bus.

However, this bus is not an ESB, but an in-memory event server.

Variety of Frameworks and Tools for Rules and Event Correlation

- Open Source
  - Drools
  - Esper
  - WSO2 CEP
  - …



- TIBCO
  - CEP: BusinessEvents
  - Streaming Analytics: StreamBase
  - Live Datamart

- Other products
  - Oracle Event Processing, Software AG’s Apama, IBM InfoSphere Streams, …

TIBCO | Live Demo

TIBCO StreamBase

TIBCO StreamBase / Live Datamart in Action...

TIBCO™

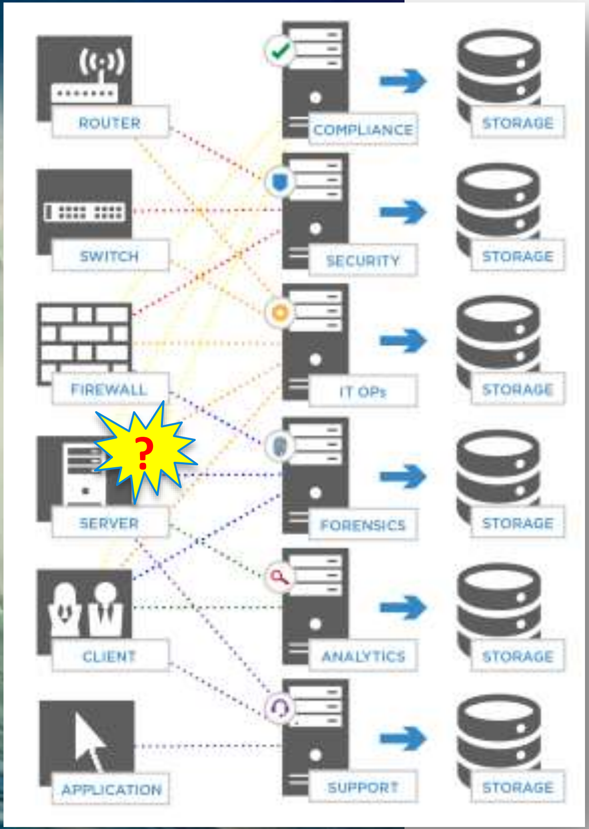
Centralized Monitoring

Applications, networks, servers, and devices generate data continuously

Growing volumes of big data add costs, complexity, and risk

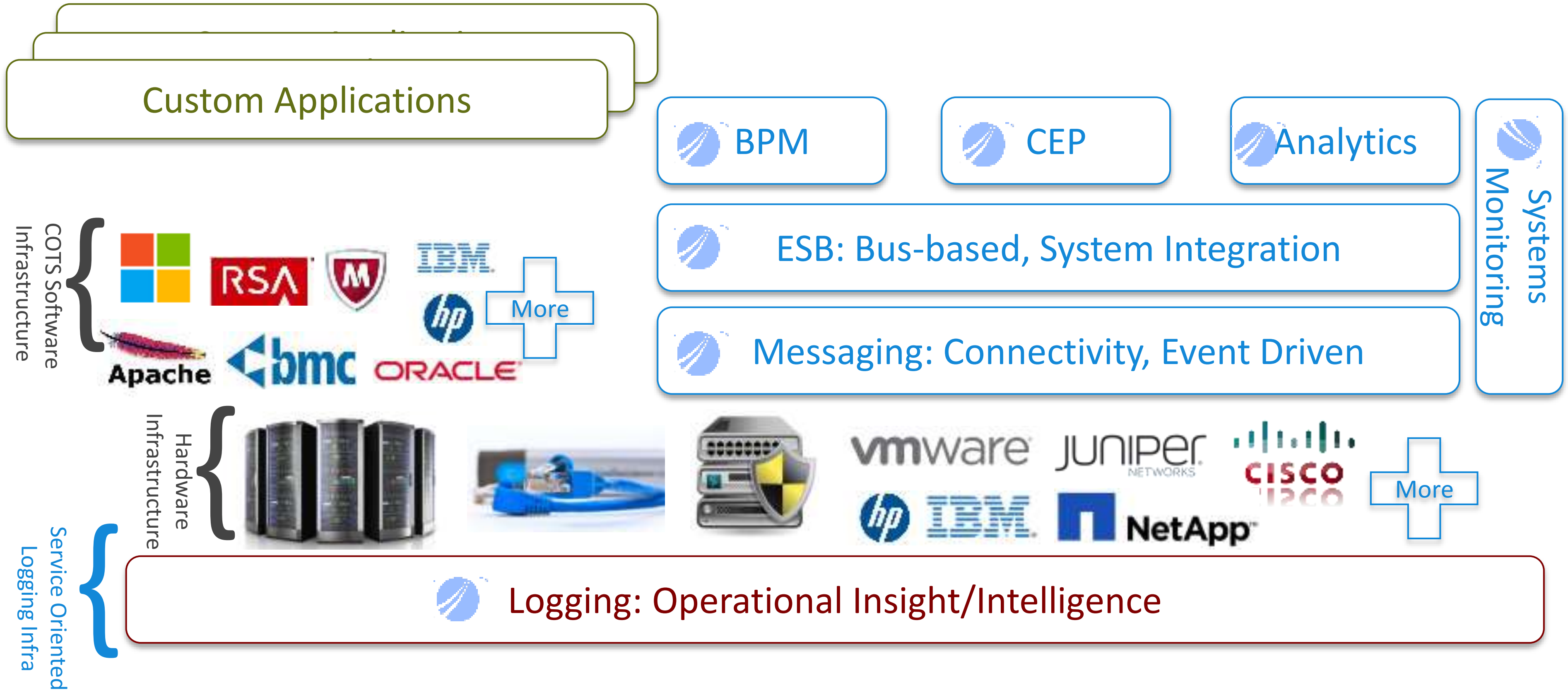
Separate, disparate and even conflicting systems can:

- Open security holes
- Jeopardize compliance
- Disrupt operations
- Impede trouble-shooting



- “Last year at this time we were processing about 20 Billion logs per day. Today that number is 54 Billion.” –Leading MSSP
- “Most of the time a client comes to us with an incident, when we try to gather the logs we discover the customer doesn’t have them and never had a policy in place to store or retain them”
- ‘Our problems are not lack of access to data, but understanding them. [Big Data] is very useful…but I have to understand it, and that's the problem."

Log Analytics in an Enterprise Architecture

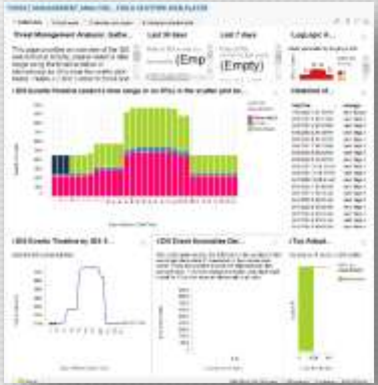

Custom Applications
BPM
CEP
Analytics
Systems Monitoring
ESB: Bus-based, System Integration
Messaging: Connectivity, Event Driven
COTS Software Infrastructure
RSA
IBM
hp
More
Apache
bmc
ORACLE
Hardware Infrastructure
vmware
JUNIPER NETWORKS
CISCO
hp
IBM
NetApp
More
Service Oriented Logging Infra
Logging: Operational Insight/Intelligence

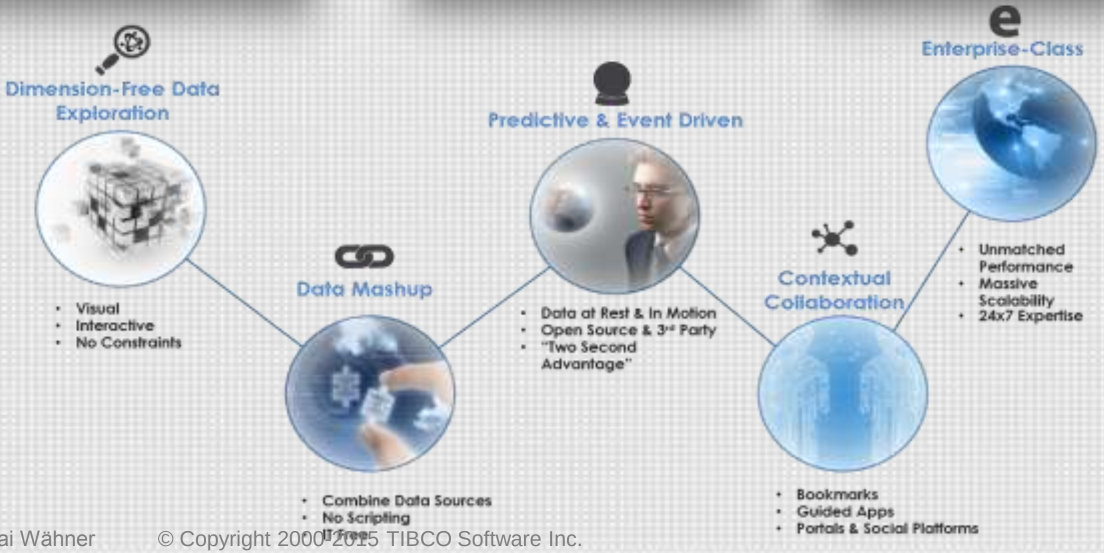
TIBCO™ | Log Analytics – Market Overview

- ELK Stack (Elasticsearch, Logstash, Kibana)
  - Open Source frameworks, coding required
- TIBCO LogLogic
  - Part of complete middleware stack, easy configuration and usage, Software or Appliance
- Splunk
  - Market leader, most complete feature list, expensive, complex architecture, turns off if volume limit reached
- HP ArcSight
  - Powerful SIEM solution, very complex configuration
- Sumo Logic
  - Small vendor, focused on log analytics
- IBM et al
  - also offer a product for Log Analytics, of course! ☺

LogLogic: Monitor, Analyze and Predict Threats and Opportunities

- Visual Log Analytics for Any Use Case
  - User Authentication Activity Tracking
  - Performance, Firewall, Network Traffic Analysis
  - Threat Management
  - Data Enrichment
- Self-Service Discovery
- Build your own dashboards
- Insight into Action
  - Quickly uncover unknown relationships, trends, and anomalies through ad-hoc query and filtering





TIBCO | Live Demo

LogLogic

TIBCO LogLogic in Action...

TIBCO™ | Agenda

- Definition of a Microservice
- Architecture Requirements
- Concepts for Microservices
- Frameworks and Tools
- **Getting Started**

TIBCO | How to continue on your Microservices Path?

1) Choose the Microservice concepts you need!

2) Think about your architecture requirements!

3) Evaluate your short list regarding features, usability, time-to-market and TCO!

4) Try out the tools by yourself within the proof of concept!

5) Choose the right tool for the right job!

TIBCO™ | Best Practices for your Microservices Path

No big bang
Start small
Keep it simple
Only if added value
Not everywhere
Not everything
Not just technologies
Organizational changes needed

Did you get the Key Message?

TIBCO | Key Messages

- Integration is key for success of Microservices! ✓
- Real time event correlation is the game changer! ✓
- TCO and Time-to-Market are major aspects for tool selection! ✓

Questions?

Kai Wähner
kwaehner@tibco.com
@KaiWaehner
www.kai-waehner.de
LinkedIn / Xing → Please connect!