31.8.–3.9.2015
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

Moderne Microservices Architekturen

Das Ende des Enterprise Service Bus (ESB)?

Kai Wähner

TIBCO

@KaiWaehner

www.kai-waehner.de

Xing / LinkedIn

TIBCO | Key Messages

- Microservices = SOA done right!
- Integration is key for success – the product name does not matter!
- Real time event correlation is the game changer!

TIBCO™ | Agenda

- Digitalization
- Enterprise Service Bus
- Microservices
- Architecture and Requirements
- Challenges

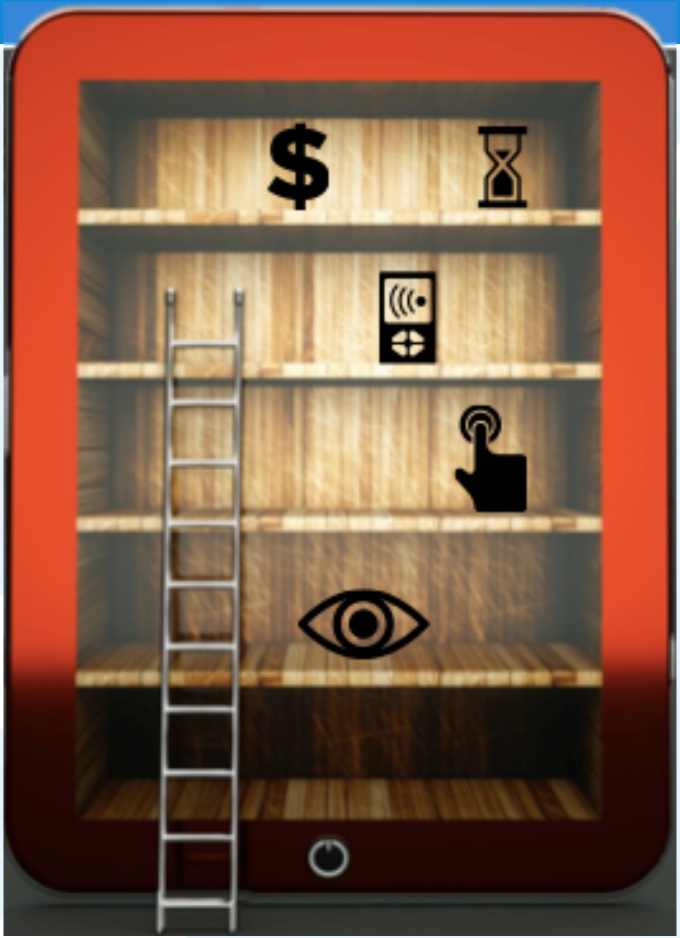
TIBCO™ | Agenda

- Digitalization
- Enterprise Service Bus
- Microservices
- Architecture and Requirements
- Challenges

Everything Generates More Data

New Applications, New Requirements

- **Purposeful**: Users are looking for tools not toolboxes.
- **Adaptable**: Similar application services can be consumed via a variety of channels, in a variety of contexts.
- **Sustainable**: Applications services need to support user experience by combining performance and flexibility.



Too Slow, Too Complex

"Applications created today using the good-old traditional architecture will be a business-constraining legacy before they are completed."

Yefim Natis, Software Defined Applications Webinar, May 8th

Agenda

- Digitalization
- **Enterprise Service Bus**
- Microservices
- Architecture and Requirements
- Challenges

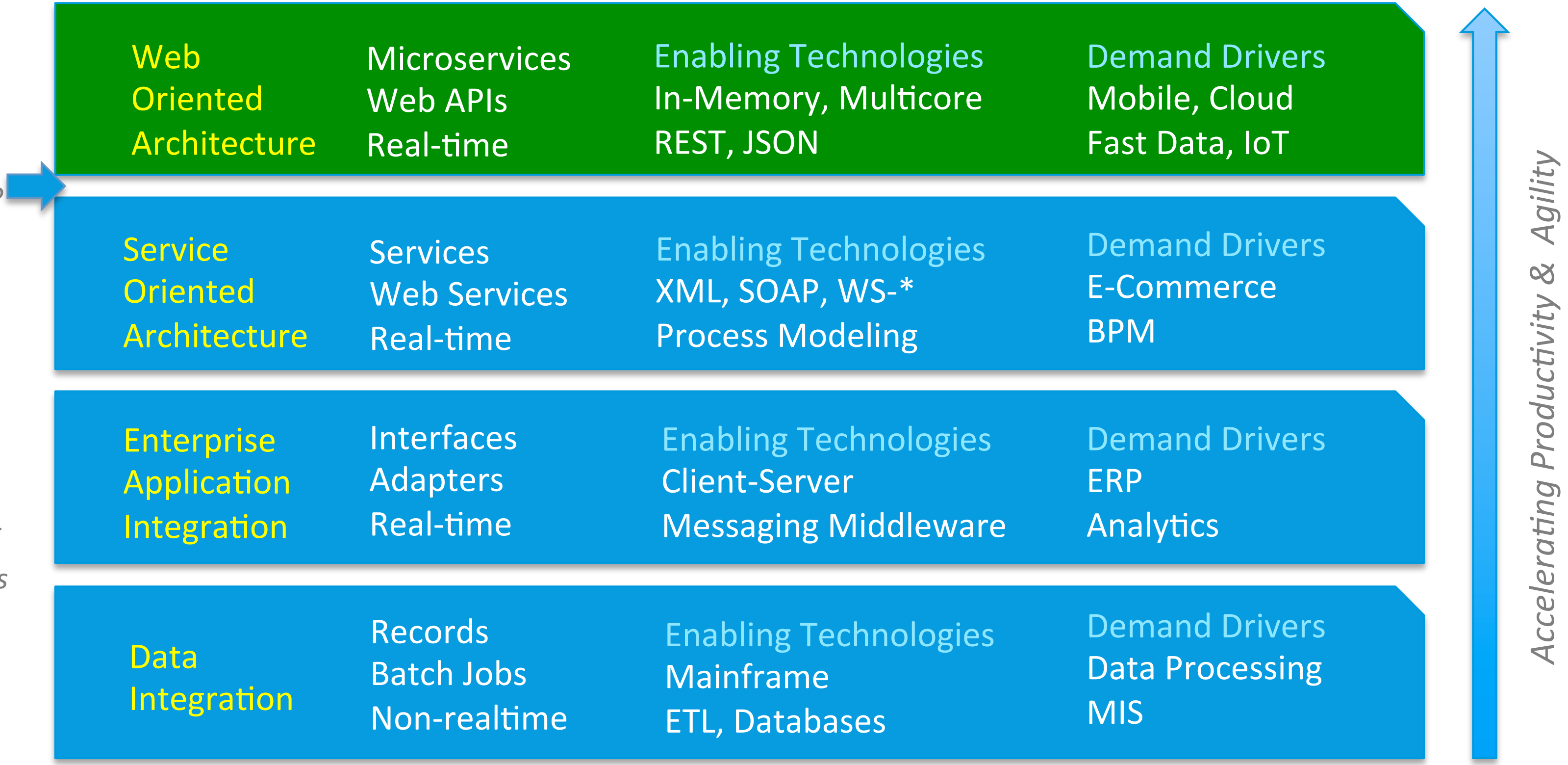
Integration is key for success!

Integration will get even more important in the future than it is today!

The number of different data sources and technologies increases even more than in the past

- CRM, ERP, Host, B2B, etc. will not disappear
- DWH, Hadoop cluster, event / streaming server, In-Memory DB – all of them have to communicate
- Cloud, Mobile, APIs, Big Data, Internet of Things are no option, but our future!
- EVERYTHING HAS TO BE INTEGRATED!

Evolution of Integration



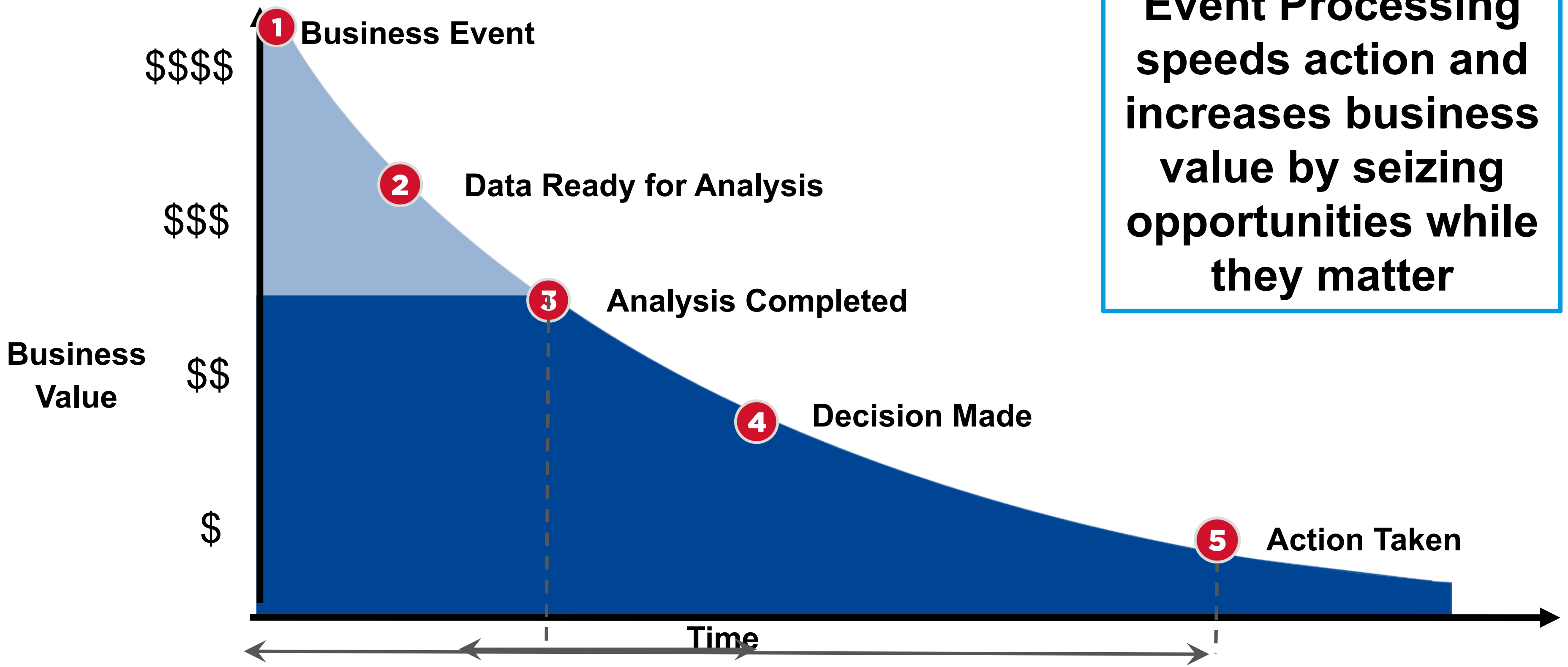
Are we there yet?

Web Oriented Architecture	Microservices Web APIs Real-time	Enabling Technologies In-Memory, Multicore REST, JSON	Demand Drivers Mobile, Cloud Fast Data, IoT
Service Oriented Architecture	Services Web Services Real-time	Enabling Technologies XML, SOAP, WS-* Process Modeling	Demand Drivers E-Commerce BPM
Enterprise Application Integration	Interfaces Adapters Real-time	Enabling Technologies Client-Server Messaging Middleware	Demand Drivers ERP Analytics
Data Integration	Records Batch Jobs Non-realtime	Enabling Technologies Mainframe ETL, Databases	Demand Drivers Data Processing MIS

Level-Up by utilizing the lessons, assets and practices of the previous Level

Accelerating Productivity & Agility

Acting in Real Time gets more and more important!



Event Processing speeds action and increases business value by seizing opportunities while they matter

“Microservices = Death of the ESB?” by Kai Wähner

Branding of Integration Software

- Keywords PAST:

  Application Integration, EAI, Broker, Application, Integration, Enterprise, Hub and Spoke, Backbone, Scalability, Platform, Batch

- Keywords PRESENT:

  Service Integration, Bus, SOA, Service, ESB, Flexibility, Distribution, Events, EDA, Real Time, Event Correlation, Open, Standards, Extensibility

- Keywords FUTURE:

  Integration of Everything, Cloud, IoT, Gateway, Microservice, API, Public Independence, Continuous Delivery, Self-Service, Prediction, In-Memory

TIBCO™ | Branding of Integration Software

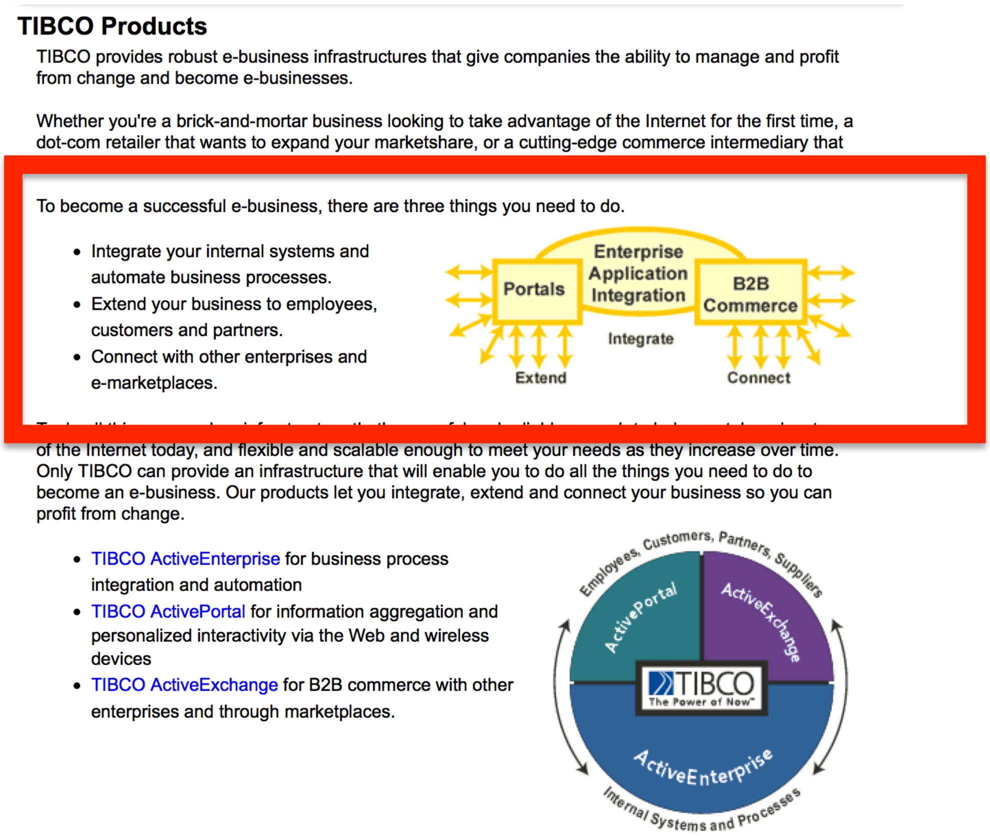
TIBCO

offers middleware for mission-critical real time

Integration and Event Processing

for 20+ Years…

TIBCO Website (Year 2000)



TIBCO Products

TIBCO provides robust e-business infrastructures that give companies the ability to manage and profit from change and become e-businesses.

Whether you're a brick-and-mortar business looking to take advantage of the Internet for the first time, a dot-com retailer that wants to expand your marketshare, or a cutting-edge commerce intermediary that

To become a successful e-business, there are three things you need to do.

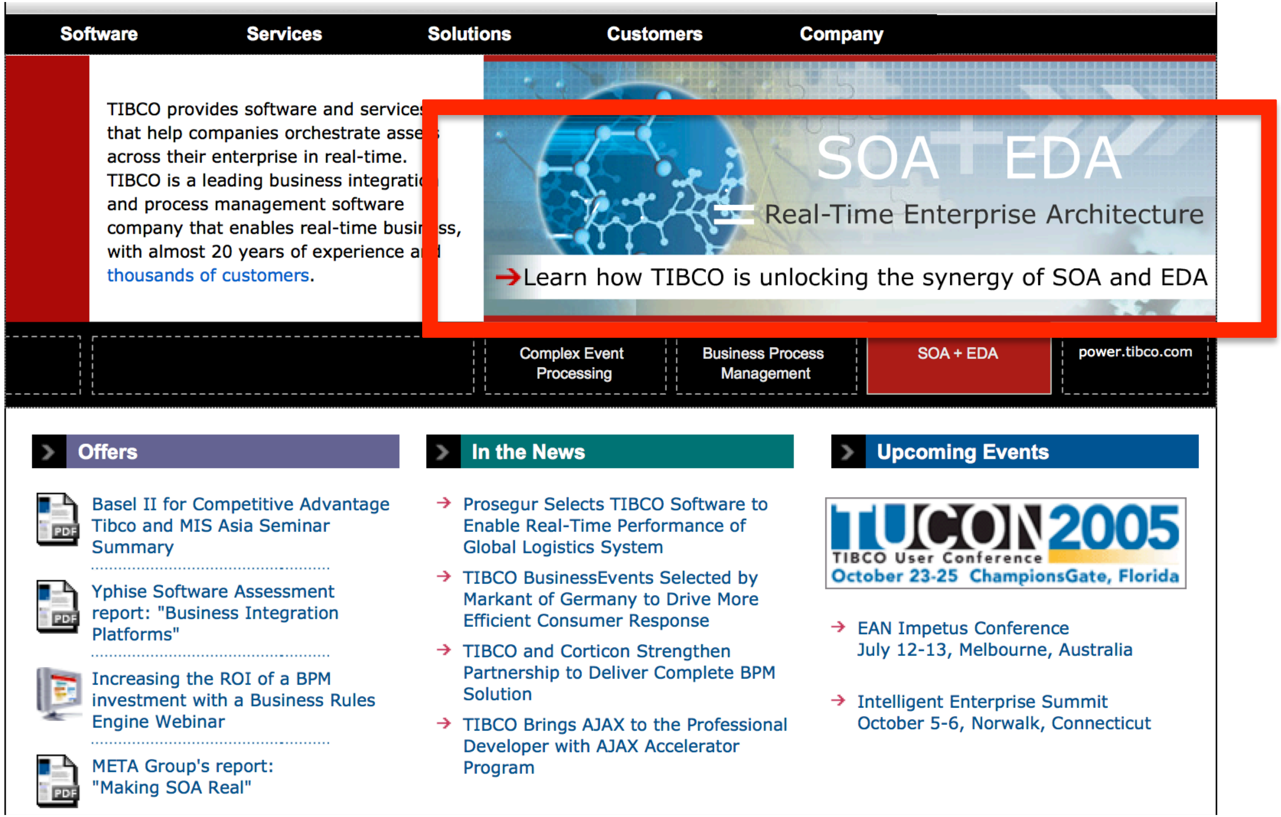
- Integrate your internal systems and automate business processes.
- Extend your business to employees, customers and partners.
- Connect with other enterprises and e-marketplaces.

of the Internet today, and flexible and scalable enough to meet your needs as they increase over time. Only TIBCO can provide an infrastructure that will enable you to do all the things you need to do to become an e-business. Our products let you integrate, extend and connect your business so you can profit from change.

- TIBCO ActiveEnterprise for business process integration and automation
- TIBCO ActivePortal for information aggregation and personalized interactivity via the Web and wireless devices
- TIBCO ActiveExchange for B2B commerce with other enterprises and through marketplaces.

Employees, Customers, Partners, Suppliers
ActivePortal
ActiveExchange
TIBCO
The Power of Now™
ActiveEnterprise
Internal Systems and Processes

TIBCO Website (Year 2005)



Software | Services | Solutions | Customers | Company

TIBCO provides software and services that help companies orchestrate asse across their enterprise in real-time. TIBCO is a leading business integratio and process management software company that enables real-time busi ss, with almost 20 years of experience a thousands of customers.

SOA + EDA = Real-Time Enterprise Architecture

→Learn how TIBCO is unlocking the synergy of SOA and EDA

Offers

- Basel II for Competitive Advantage Tibco and MIS Asia Seminar Summary
- Yphise Software Assessment report: "Business Integration Platforms"
- Increasing the ROI of a BPM investment with a Business Rules Engine Webinar
- META Group's report: "Making SOA Real"

In the News

→ Prosegur Selects TIBCO Software to Enable Real-Time Performance of Global Logistics System

→ TIBCO BusinessEvents Selected by Markant of Germany to Drive More Efficient Consumer Response

→ TIBCO and Corticon Strengthen Partnership to Deliver Complete BPM Solution

→ TIBCO Brings AJAX to the Professional Developer with AJAX Accelerator Program

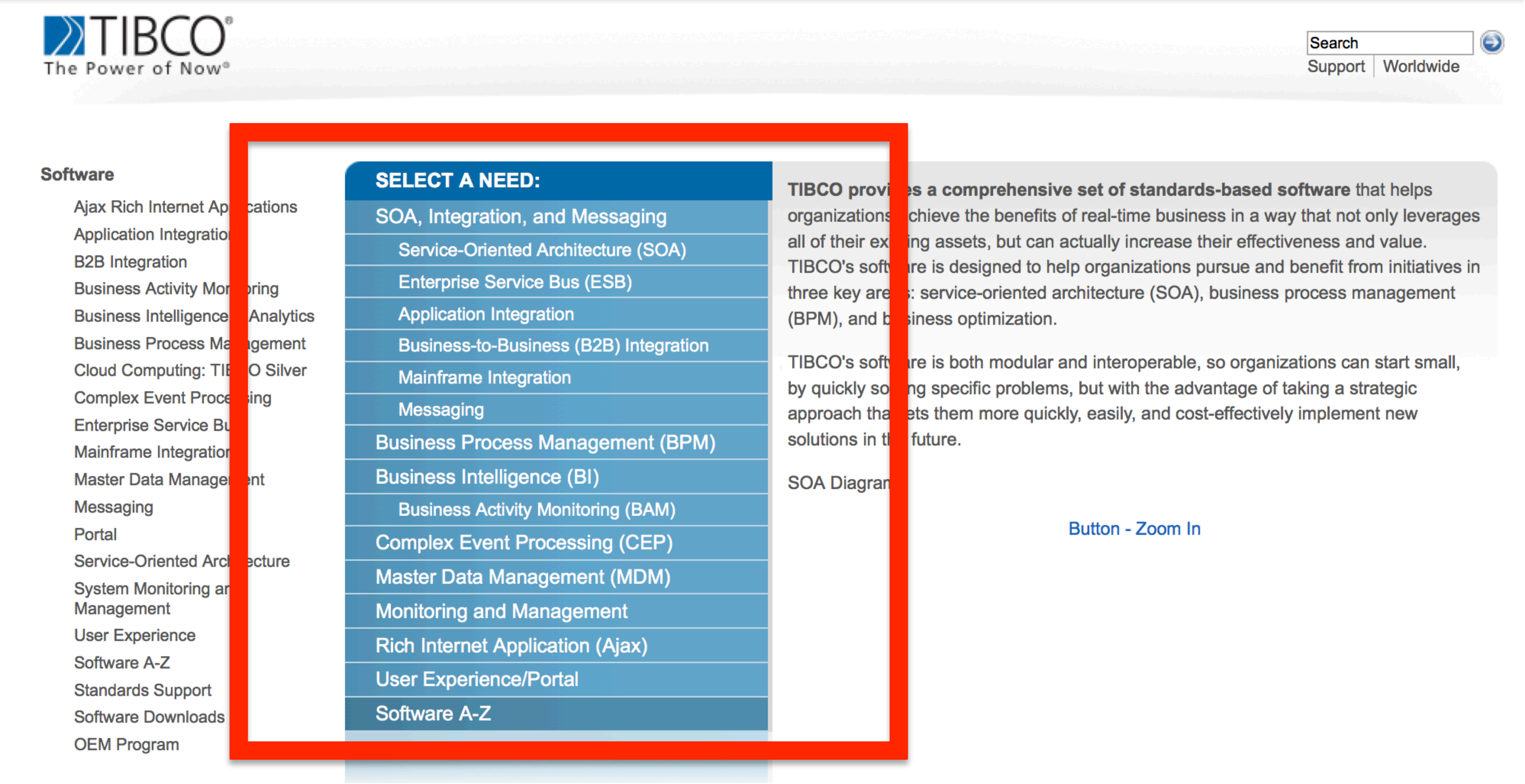
Upcoming Events

TUCON2005
TIBCO User Conference
October 23-25 ChampionsGate, Florida

→ EAN Impetus Conference
July 12-13, Melbourne, Australia

→ Intelligent Enterprise Summit
October 5-6, Norwalk, Connecticut

TIBCO Website (Year 2010)



Software

- Ajax Rich Internet Ap cations
- Application Integratio
- B2B Integration
- Business Activity Mo oring
- Business Intelligence Analytics
- Business Process Ma gement
- Cloud Computing: TI O Silver
- Complex Event Proce ing
- Enterprise Service Bu
- Mainframe Integratio
- Master Data Manage ent
- Messaging
- Portal
- Service-Oriented Arc ecture
- System Monitoring a Management
- User Experience
- Software A-Z
- Standards Support
- Software Downloads
- OEM Program

SELECT A NEED:

- SOA, Integration, and Messaging
  - Service-Oriented Architecture (SOA)
  - Enterprise Service Bus (ESB)
  - Application Integration
  - Business-to-Business (B2B) Integration
  - Mainframe Integration
  - Messaging
- Business Process Management (BPM)
- Business Intelligence (BI)
  - Business Activity Monitoring (BAM)
- Complex Event Processing (CEP)
- Master Data Management (MDM)
- Monitoring and Management
- Rich Internet Application (Ajax)
- User Experience/Portal
- Software A-Z

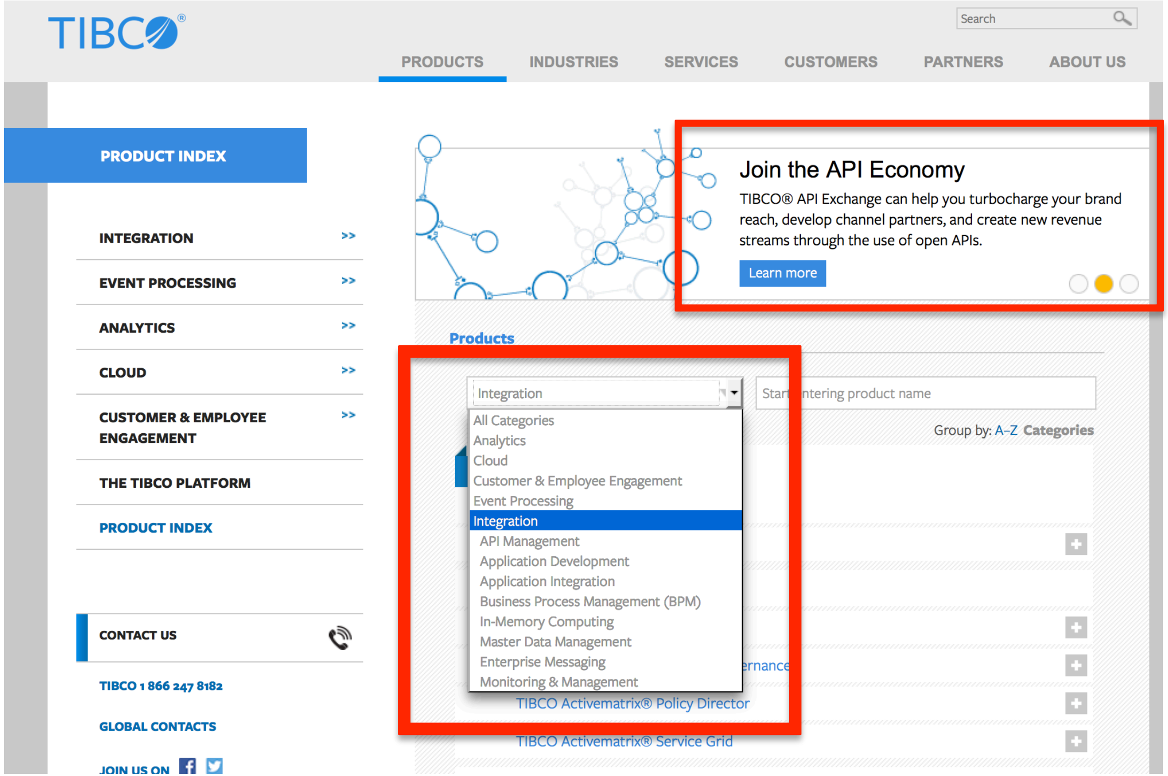
TIBCO provi es a comprehensive set of standards-based software that helps organizations chieve the benefits of real-time business in a way that not only leverages all of their ex ing assets, but can actually increase their effectiveness and value. TIBCO's soft re is designed to help organizations pursue and benefit from initiatives in three key are : service-oriented architecture (SOA), business process management (BPM), and b iness optimization.

TIBCO's soft re is both modular and interoperable, so organizations can start small, by quickly so ng specific problems, but with the advantage of taking a strategic approach tha ets them more quickly, easily, and cost-effectively implement new solutions in t future.

SOA Diagra

Button - Zoom In

TIBCO Website (Year 2015)



Join the API Economy

TIBCO® API Exchange can help you turbocharge your brand reach, develop channel partners, and create new revenue streams through the use of open APIs.

Branding of Integration Software

Same story for

IBM, Oracle, Software AG, …

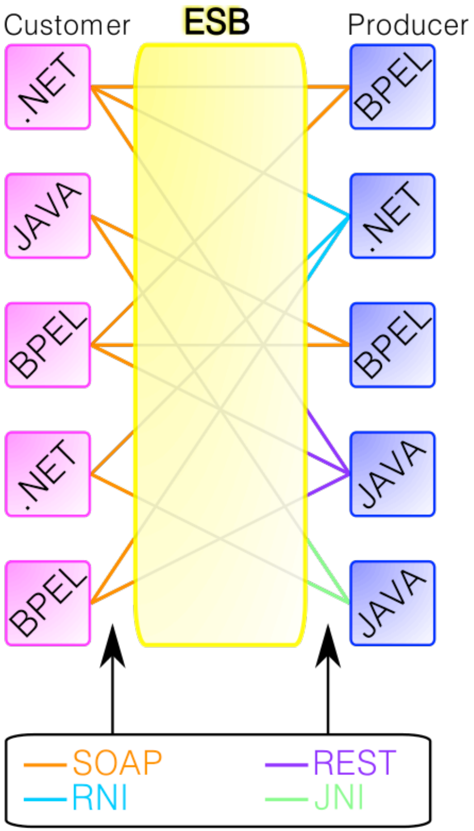
Branding of Integration Software

TIBCO BusinessWorks

(which is its integration flagship product) was

never branded ESB

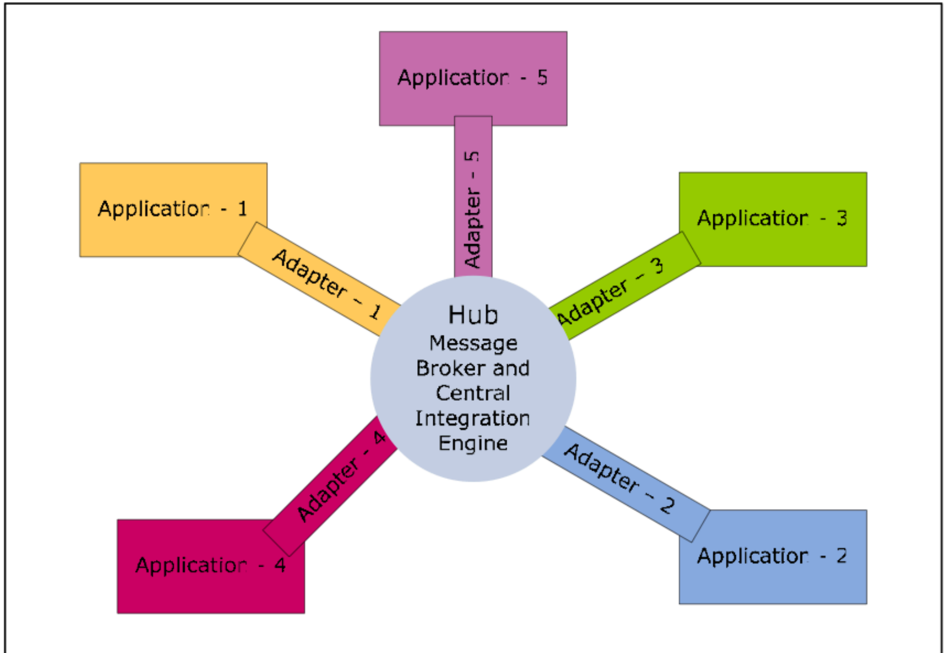
TIBCO | Enterprise Service Bus (ESB)



An enterprise service bus (ESB) is a **software architecture model** used for designing and implementing communication between mutually interacting software applications in a service-oriented architecture (SOA). Its primary use is in enterprise application integration (EAI) of heterogeneous and complex landscapes.

http://en.wikipedia.org/wiki/Enterprise_service_bus (Wikipedia, 2015)

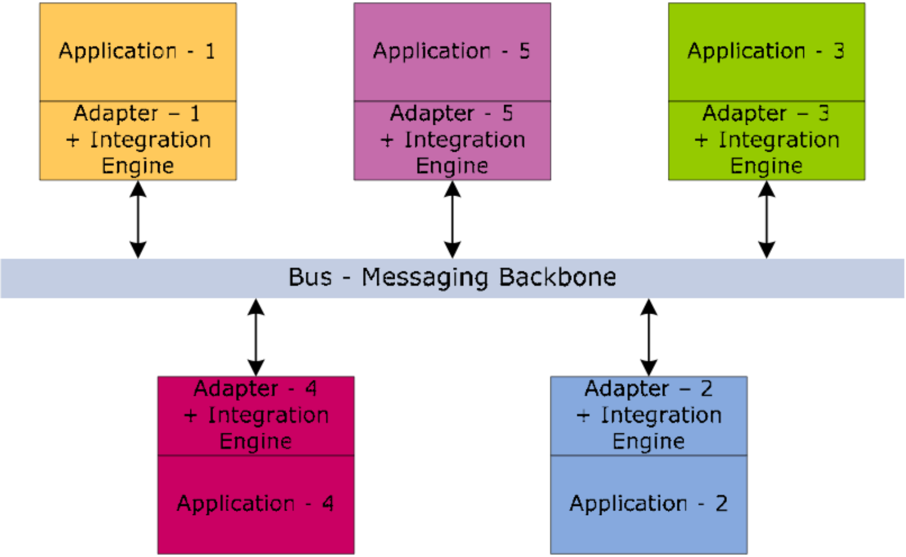
Central ESB



Hub/Spoke Architecture: Central Message broker with integration engine and adapters as spoke

http://stage.reflectsoftware.com/SOA/Enterprise%20Integration%20EAI%20vs.%20SOA%20vs.%20ESB.pdf

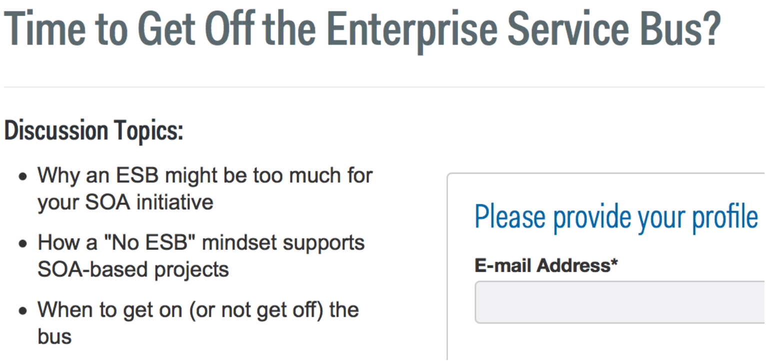
Distributed ESB


Application - 1
Adapter – 1 + Integration Engine
Application - 5
Adapter - 5 + Integration Engine
Application - 3
Adapter – 3 + Integration Engine
Bus - Messaging Backbone
Adapter - 4 + Integration Engine
Application - 4
Adapter – 2 + Integration Engine
Application - 2

Bus Architecture: Central messaging backbone and distributed integration engine at adapters

http://stage.reflectsoftware.com/SOA/Enterprise%20Integration%20EAI%20vs.%20SOA%20vs.%20ESB.pdf

Is the ESB dead? #NoESB



Time to Get Off the Enterprise Service Bus?

Discussion Topics:

- Why an ESB might be too much for your SOA initiative
- How a "No ESB" mindset supports SOA-based projects
- When to get on (or not get off) the bus

Please provide your profile

E-mail Address*

Gartner
https://www.gartner.com/user/registration/webinar?resId=2855231&commId=128383&channelId=5500&srcId=null

The Message Behind #NoESB

By Roberto Medrano December 8, 2014 2 Comments

Akana (former SOA Software)
https://blog.soa.com/noesb/

TIBCO™

Is the ESB dead? #NoESB

- Do not care about branding
- What is a “modern” ESB?
  - flexible, distributed, scalable infrastructure
  - build, deploy and monitor any kind of (micro)services in an agile, efficient way with open standards
  - Development and deployment can be done on-premise, in the cloud, or a mixture of both
  - Be aware of re-branded central EAI brokers with old code base and new name. Watch out for API-only platforms, which re-implement ESB features.
- What to use an ESB for?
  - Integration, orchestration, routing, (some kinds of) event processing / correlation / business activity monitoring
  - API and REST are great. However, have you ever used a mature and powerful SAP connector? Or what about Internet of Things – it needs messaging (WebSockets, MQTT, AMQP, …)?
  - You can also build business applications via (micro)services, which implement your requirements and solve your business problems
  - Deploy these services independently from each other with a standardized interface to a scalable runtime platform – automatically
  - The services are decoupled and scale linearly across commodity hardware
  - Think of an ESB as a “service delivery platform”, not just an integration platform

Agenda

- Digitalization
- Enterprise Service Bus
- **Microservices**
- Architecture and Requirements
- Challenges

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….

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
- 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™ | Agenda

- Digitalization
- Enterprise Service Bus
- Microservices
- **Architecture and Requirements**
- Challenges

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

#1: Services Contract

Service Contracts

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

Other developers can easily access this information

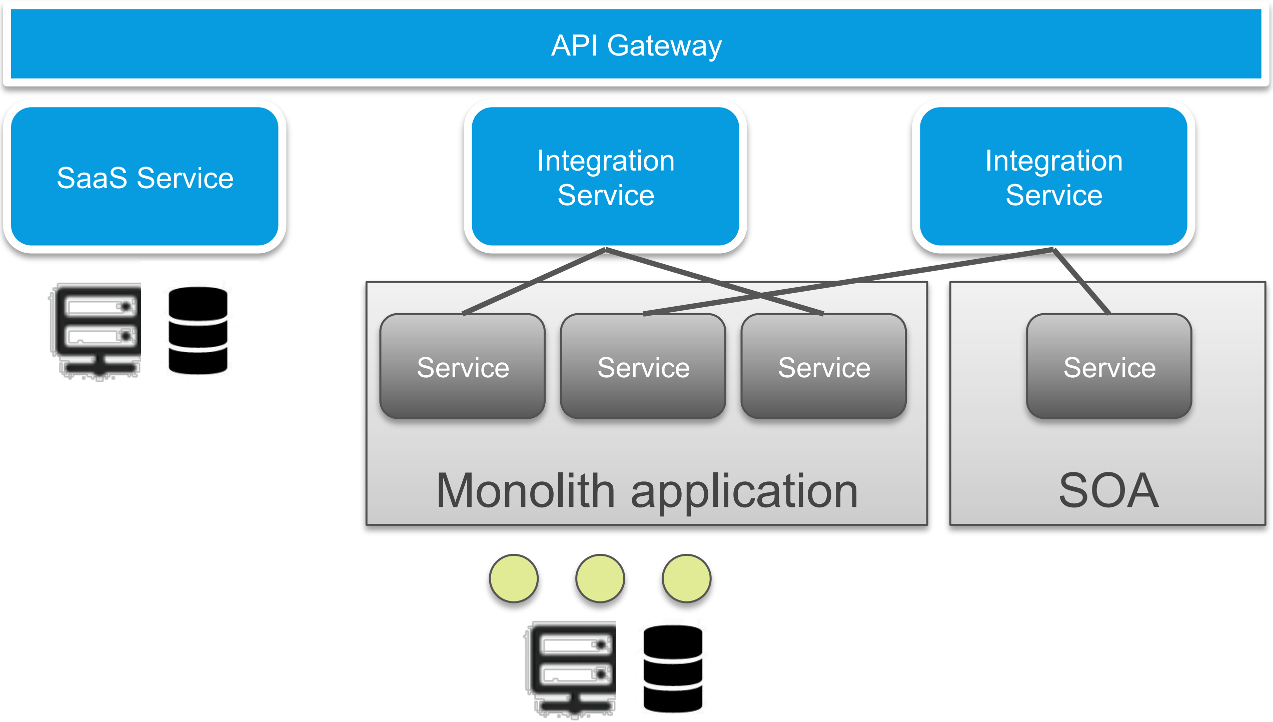
Service contracts, and the ability for developers to discover them, serve that purpose.

TIBCO | Technologies for (Micro)Service Contracts

- Examples: Java Interface, JMS, 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!

#2: Exposing new and existing Microservices

Services come in various forms



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

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”

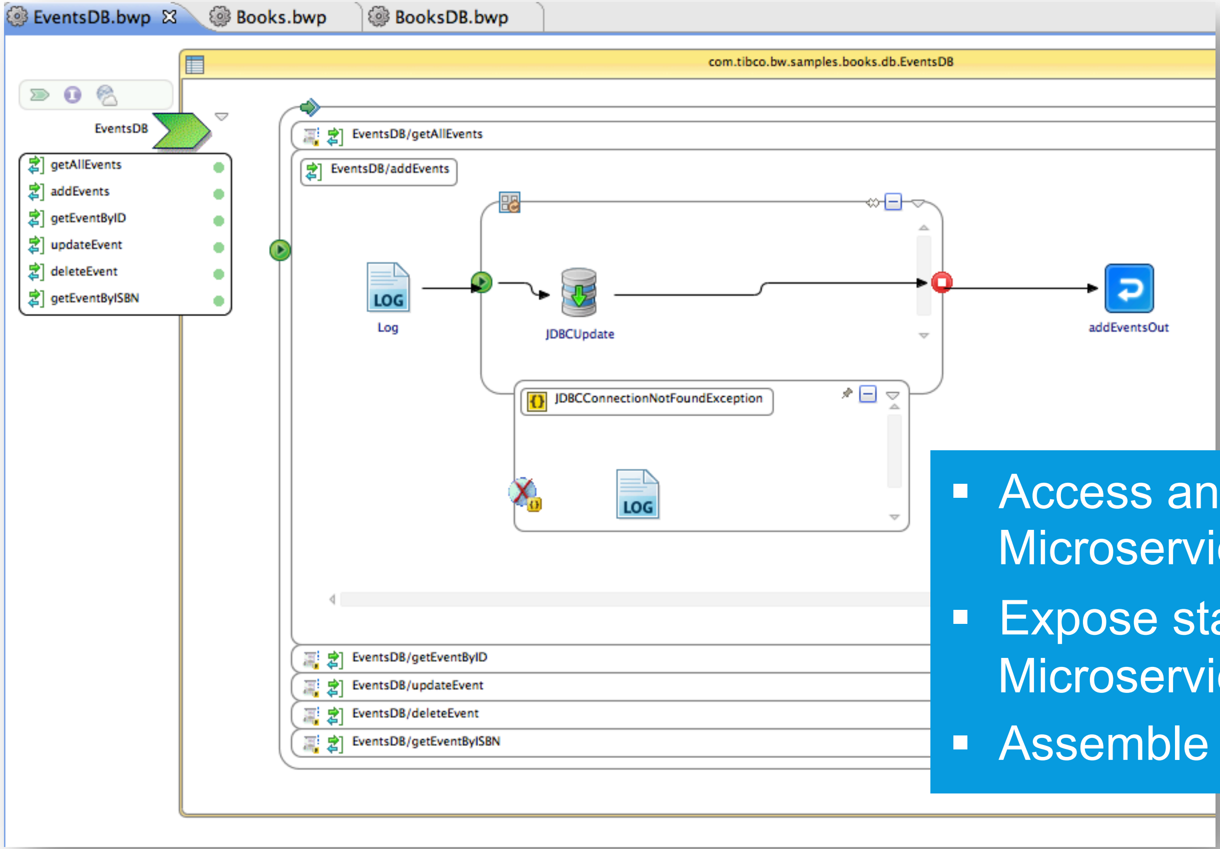
Agreed!

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?

TIBCO™

Integration as foundation of Microservices

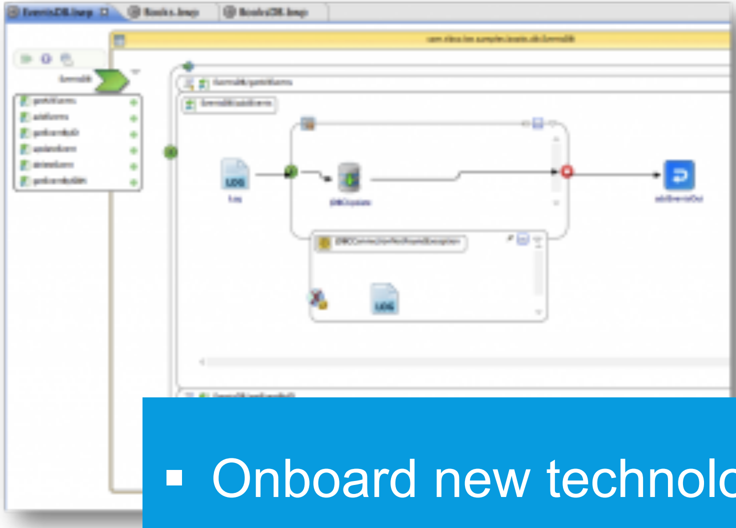


- Access any data to use in Microservices
- Expose standard transport from Microservices
- Assemble new Microservices

TIBCO | Leverage any technology to create Microservices

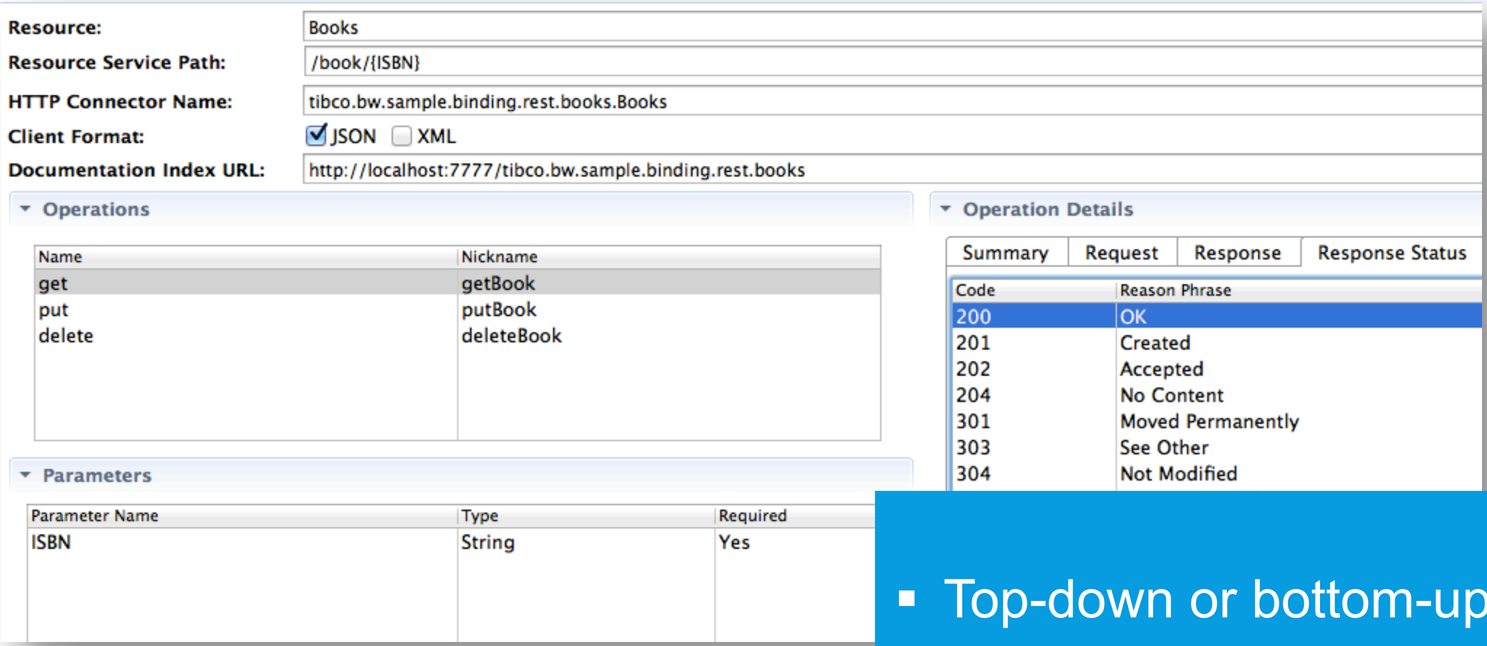
Abstract complex APIs using:

- 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.



- Onboard new technologies
- New channels
- New data sources

Create REST APIs service quickly



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

Operation Details

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

Parameters

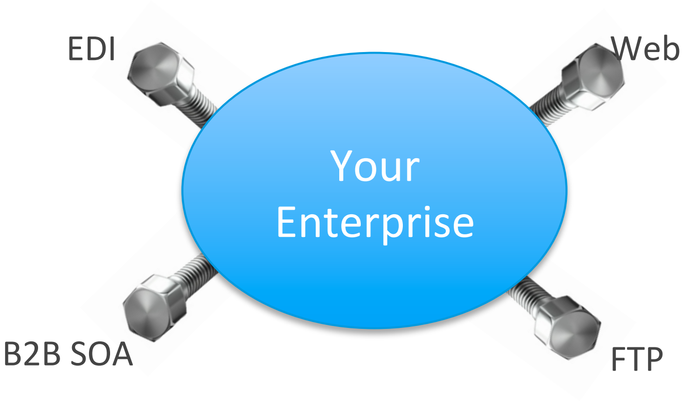
Parameter Name	Type	Required
ISBN	String	Yes

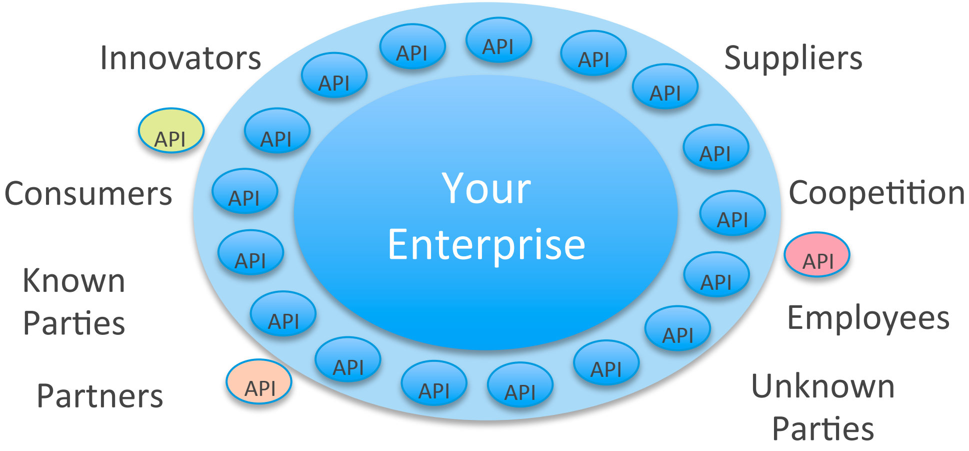
- Top-down or bottom-up modeling
- Automatic docs and testing web UI

#3: Discovery of Services

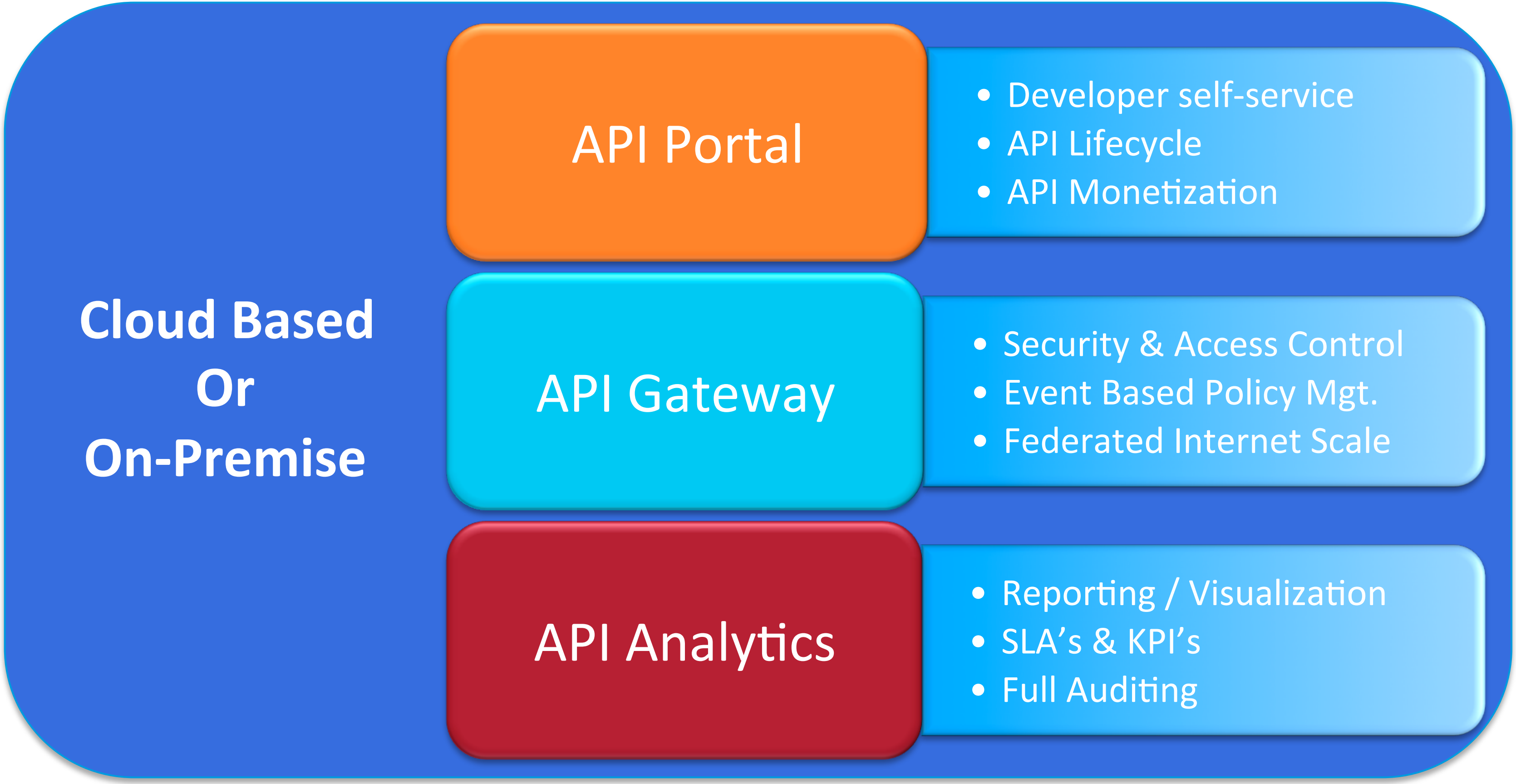
The new “Open API” Economy

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





TIBCO | API Management



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

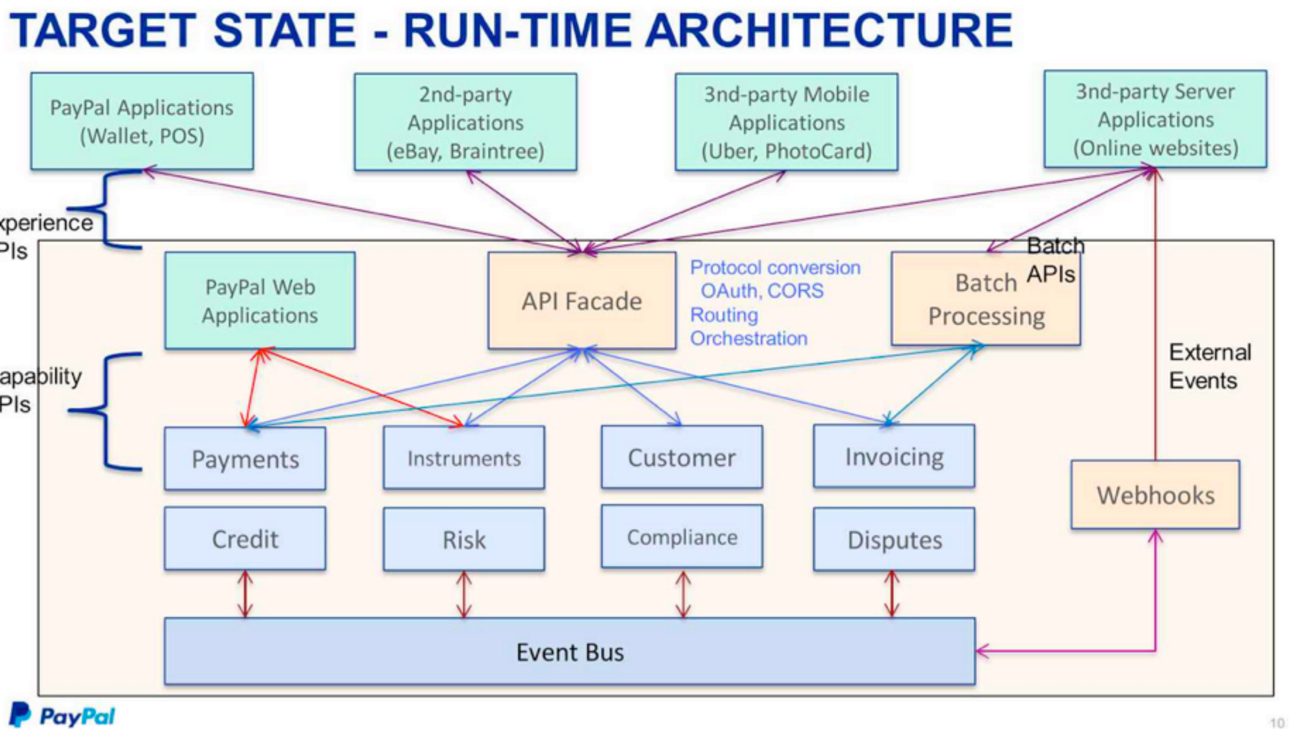
Real World Use Cases for Open API

- Paypal (eCommerce consumer) ➔ Pay everything with the same online payment service in a secure, but also very easy way

- Amazon Web Services (IT infrastructure) ➔ Use Amazon‘s gigantic data center in a flexible, elastic, but also very cheap way for your changing computation demands

- Domino‘s Pizza (mobile enablement) ➔ Order your next pizza from your smartphone app (includes choosing menu, using coupons, doing payment - via Paypal API for instance)

API Management at Paypal



TARGET STATE - RUN-TIME ARCHITECTURE

PayPal Applications (Wallet, POS)

2nd-party Applications (eBay, Braintree)

3nd-party Mobile Applications (Uber, PhotoCard)

3nd-party Server Applications (Online websites)

Experience APIs

PayPal Web Applications

API Facade

Protocol conversion
OAuth, CORS
Routing
Orchestration

Batch Processing

Batch APIs

External Events

Capability APIs

Payments

Instruments

Customer

Invoicing

Webhooks

Credit

Risk

Compliance

Disputes

Event Bus

PayPal

10

http://www.tibco.com/blog/2015/03/23/creating-business-value-by-example-open-api-and-api-management-at-paypal/

#4: Coordination across services

TIBCO™ |

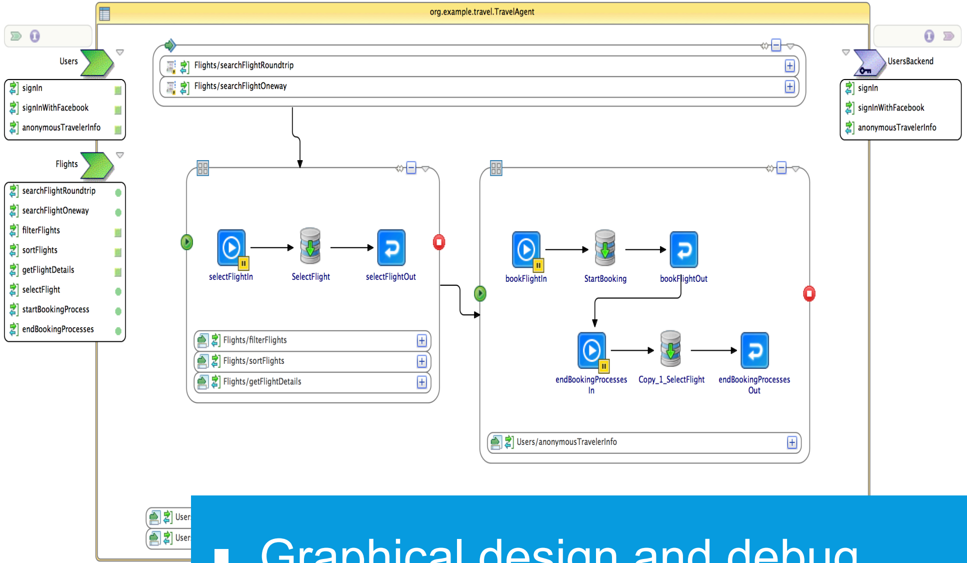
Coordination across services via NEW services ?!

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

How to coordinate?

Coordination across Services

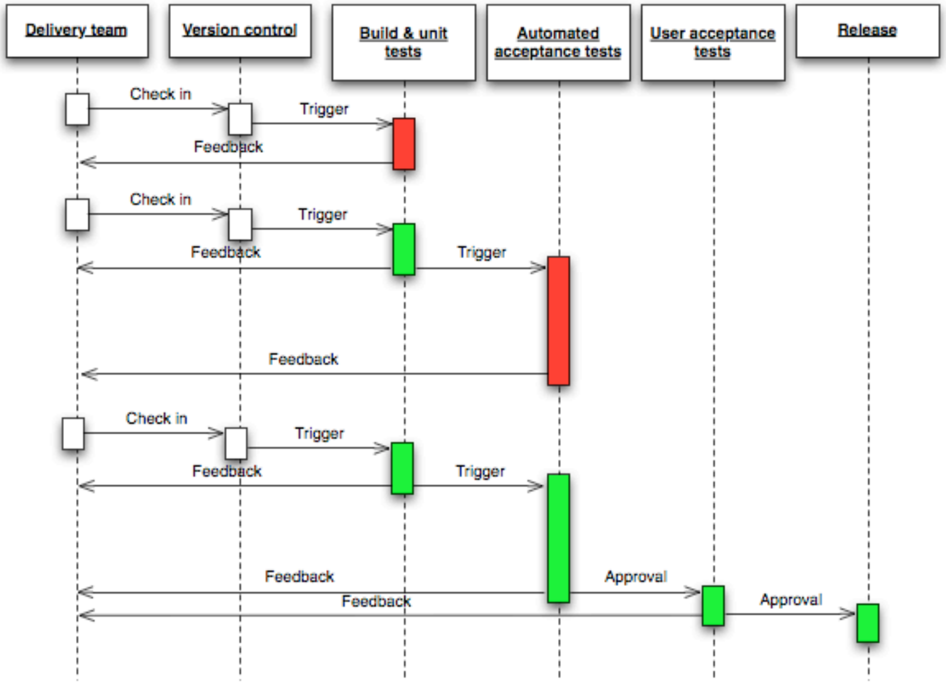
- Apps / business services are composed from Microservices
- Some Microservices can be composed to accelerate developments



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

#5: Managing complex deployments and their scalability

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

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, Vagrant, Openstack
  - Docker, Spring Boot

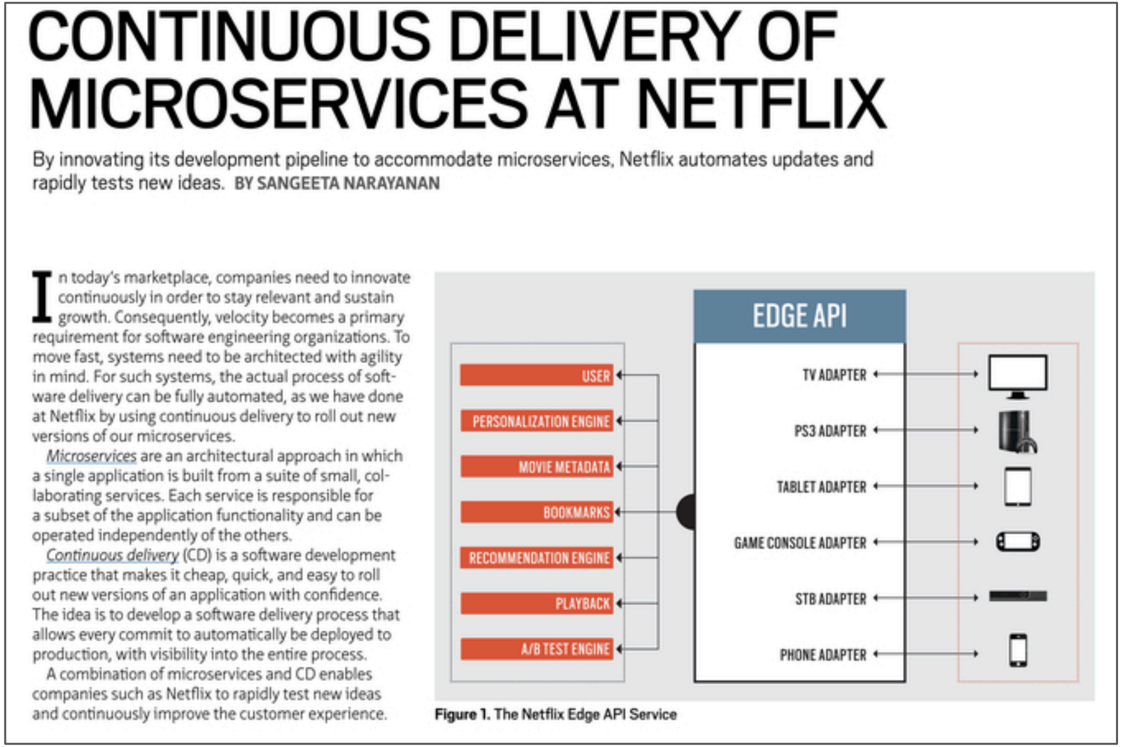
Continuous Delivery at Netflix

NETFLIX

"In today's market, companies need to innovate continuously"

"Velocity becomes a key requirement in software engineering organizations"

"Canary testing rolls out new features to a small set of end users via immutable deployments"



CONTINUOUS DELIVERY OF MICROSERVICES AT NETFLIX

By innovating its development pipeline to accommodate microservices, Netflix automates updates and rapidly tests new ideas. BY SANGEETA NARAYANAN

In today's marketplace, companies need to innovate continuously in order to stay relevant and sustain growth. Consequently, velocity becomes a primary requirement for software engineering organizations. To move fast, systems need to be architected with agility in mind. For such systems, the actual process of software delivery can be fully automated, as we have done at Netflix by using continuous delivery to roll out new versions of our microservices.

Microservices are an architectural approach in which a single application is built from a suite of small, collaborating services. Each service is responsible for a subset of the application functionality and can be operated independently of the others.

Continuous delivery (CD) is a software development practice that makes it cheap, quick, and easy to roll out new versions of an application with confidence. The idea is to develop a software delivery process that allows every commit to automatically be deployed to production, with visibility into the entire process.

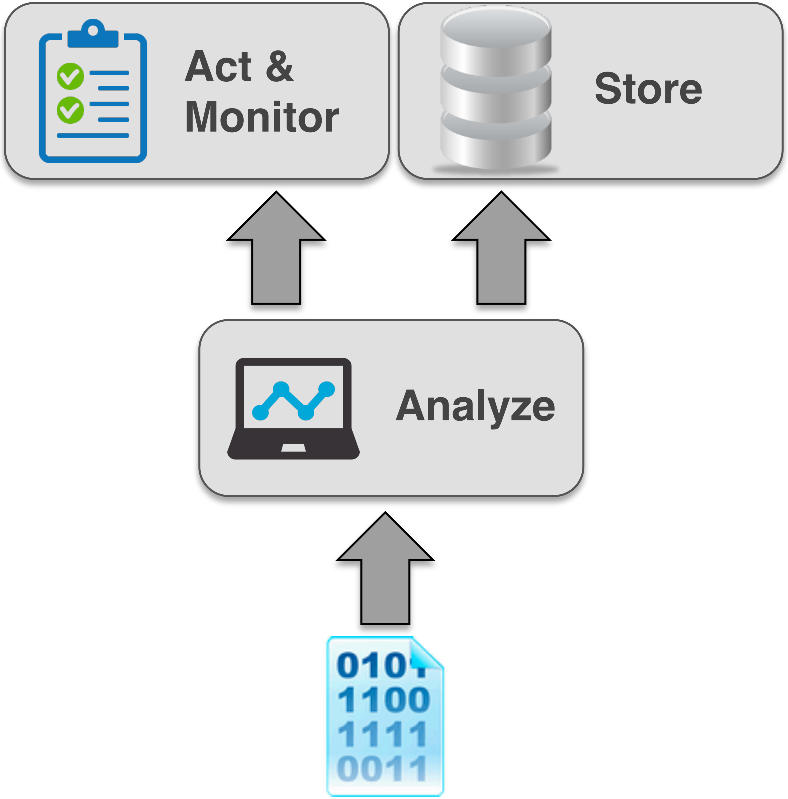
A combination of microservices and CD enables companies such as Netflix to rapidly test new ideas and continuously improve the customer experience.

Figure 1. The Netflix Edge API Service

http://www.oraclejavamagazine-digital.com/javamagazine/july_august_2015

#6: Visibility and Correlation across Services

The New Era: Fast Data Processing


Act & Monitor

Store

Analyze

- 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

TIBCO Predictive Sensor Analytics

Data Monitoring

- Motor temperature
- Motor vibration
- Current
- Intake pressure
- Intake temperature

➢ Flow

Pump Components

Electrical power cable

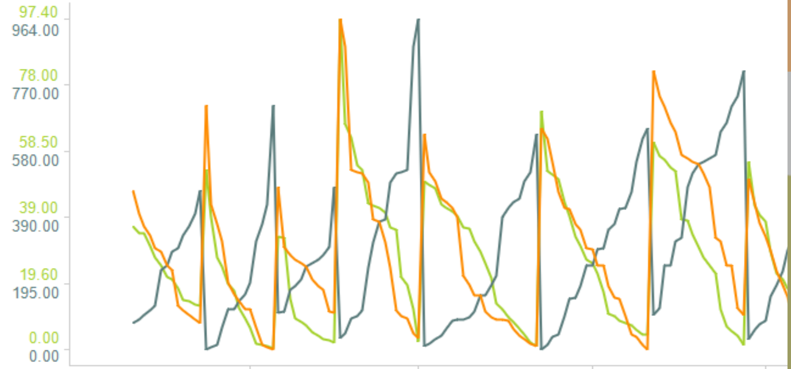
Pump

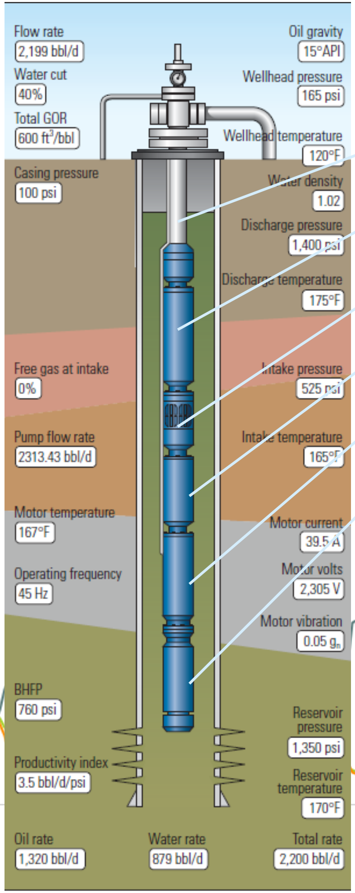
Intake

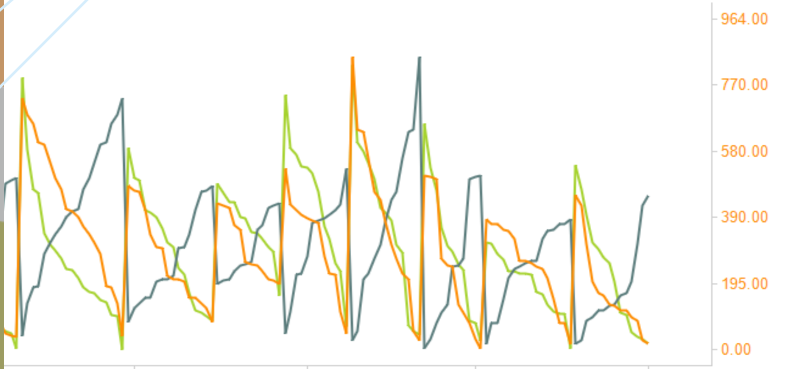
Protector

ESP motor

Pump monitoring unit

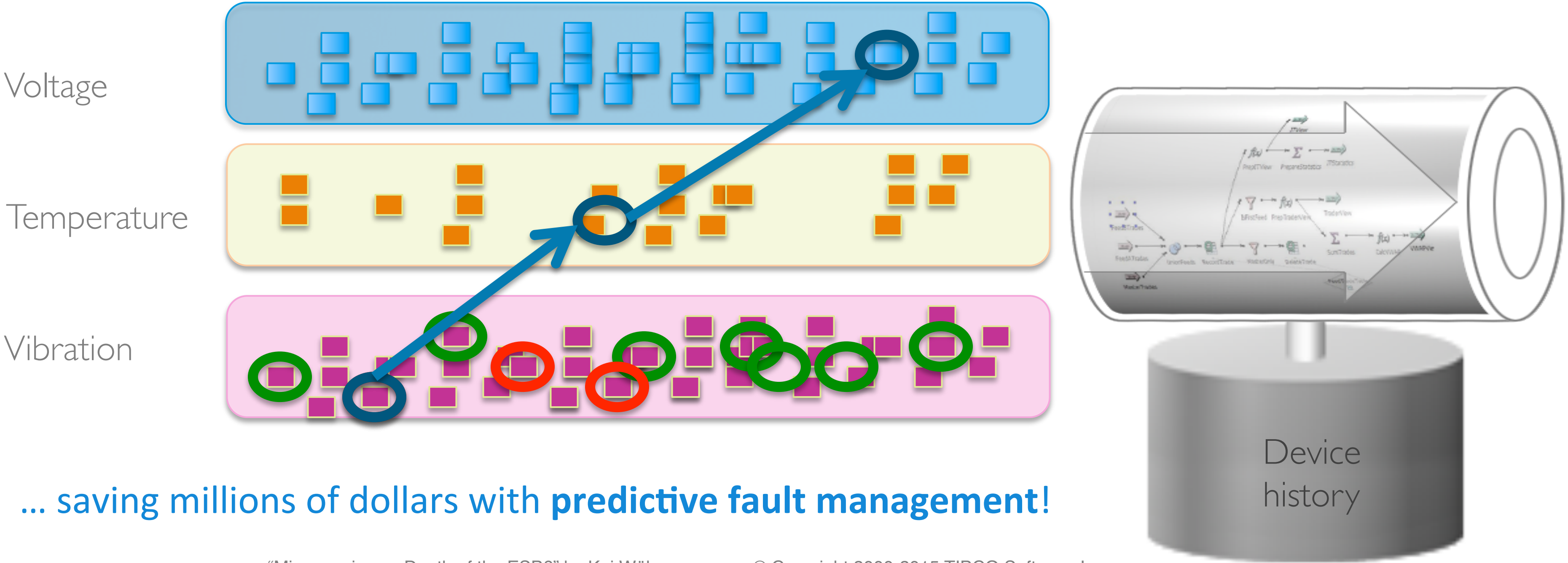






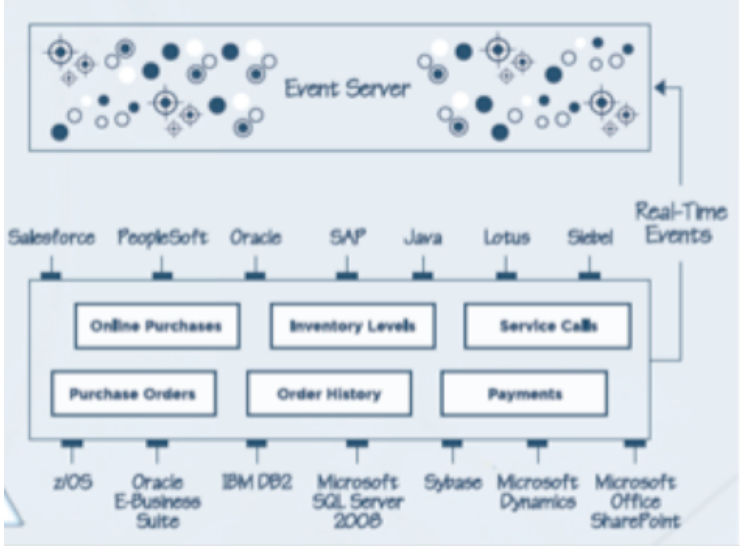
Event Processing (Correlation of Events)

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



… saving millions of dollars with **predictive fault management**!

Microservices and the need for a "Bus"?

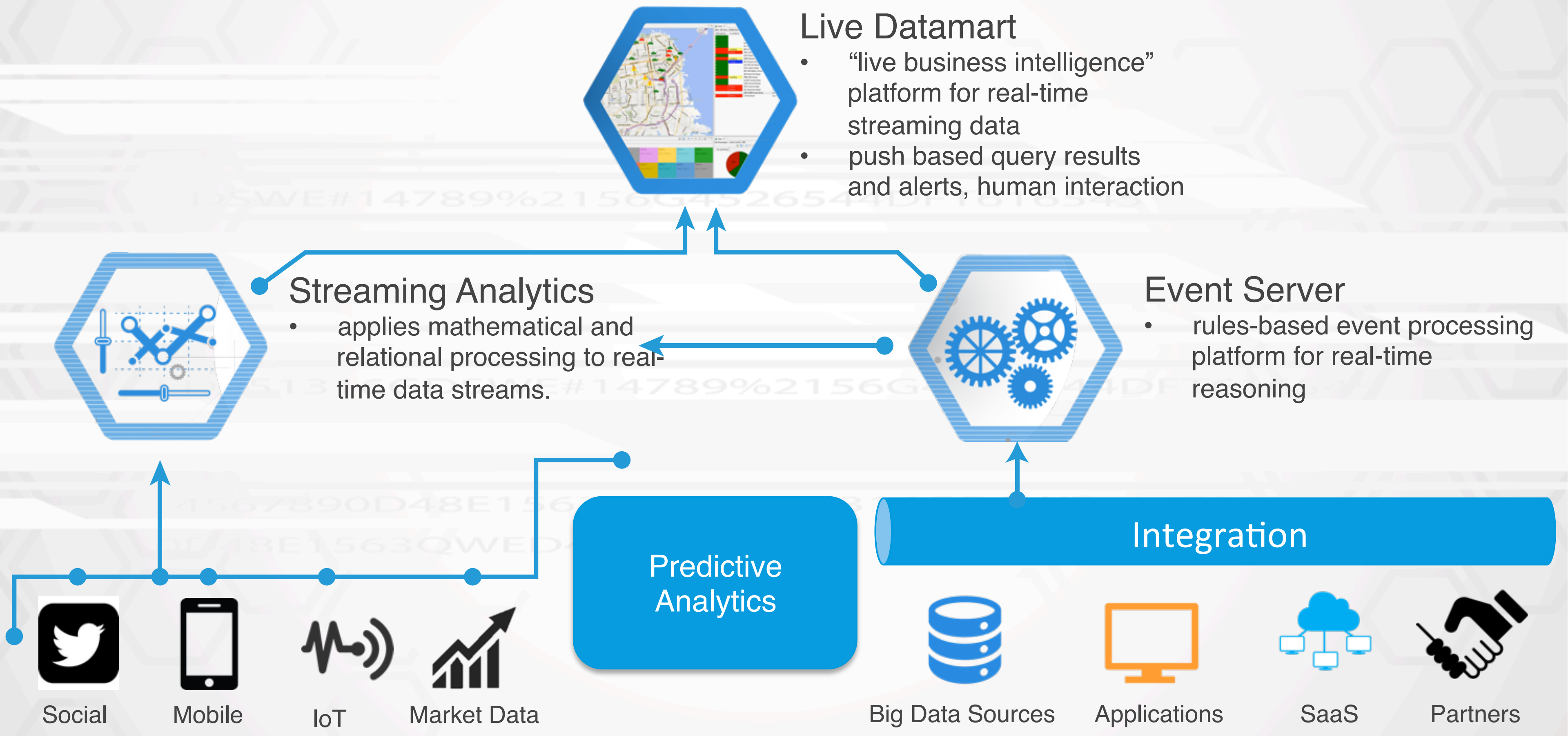


Event correlation is the requirement, where you really need a "bus".

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

TIBCO™ |

Event Processing: Stream & Event-Based



Live Datamart

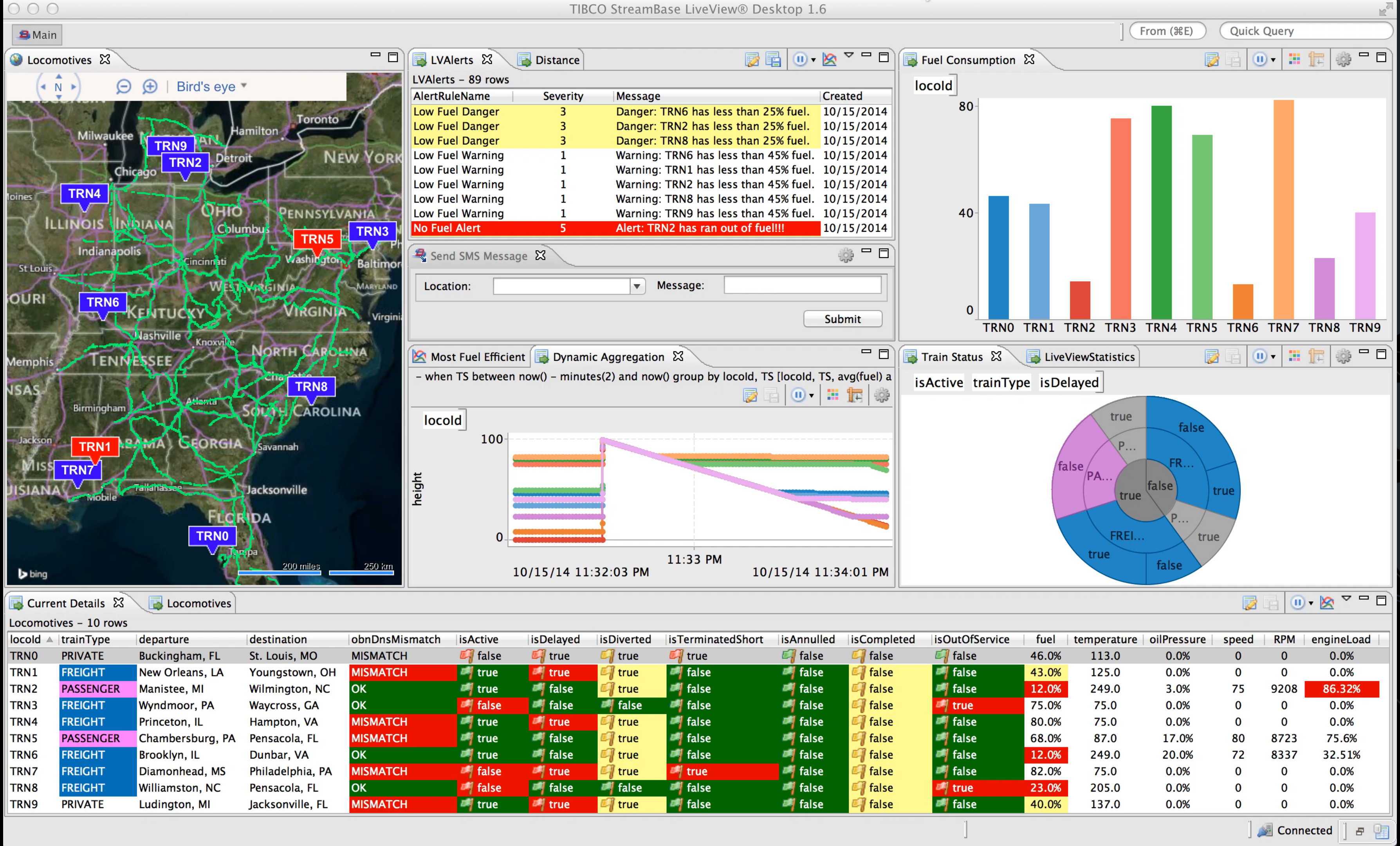
- “live business intelligence” platform for real-time streaming data
- push based query results and alerts, human interaction

Streaming Analytics

- applies mathematical and relational processing to real-time data streams.

Event Server

- rules-based event processing platform for real-time reasoning



LVAlerts – 89 rows

AlertRuleName	Severity	Message	Created
Low Fuel Danger	3	Danger: TRN6 has less than 25% fuel.	10/15/2014
Low Fuel Danger	3	Danger: TRN2 has less than 25% fuel.	10/15/2014
Low Fuel Danger	3	Danger: TRN8 has less than 25% fuel.	10/15/2014
Low Fuel Warning	1	Warning: TRN6 has less than 45% fuel.	10/15/2014
Low Fuel Warning	1	Warning: TRN1 has less than 45% fuel.	10/15/2014
Low Fuel Warning	1	Warning: TRN2 has less than 45% fuel.	10/15/2014
Low Fuel Warning	1	Warning: TRN8 has less than 45% fuel.	10/15/2014
Low Fuel Warning	1	Warning: TRN9 has less than 45% fuel.	10/15/2014
No Fuel Alert	5	Alert: TRN2 has ran out of fuel!!!	10/15/2014

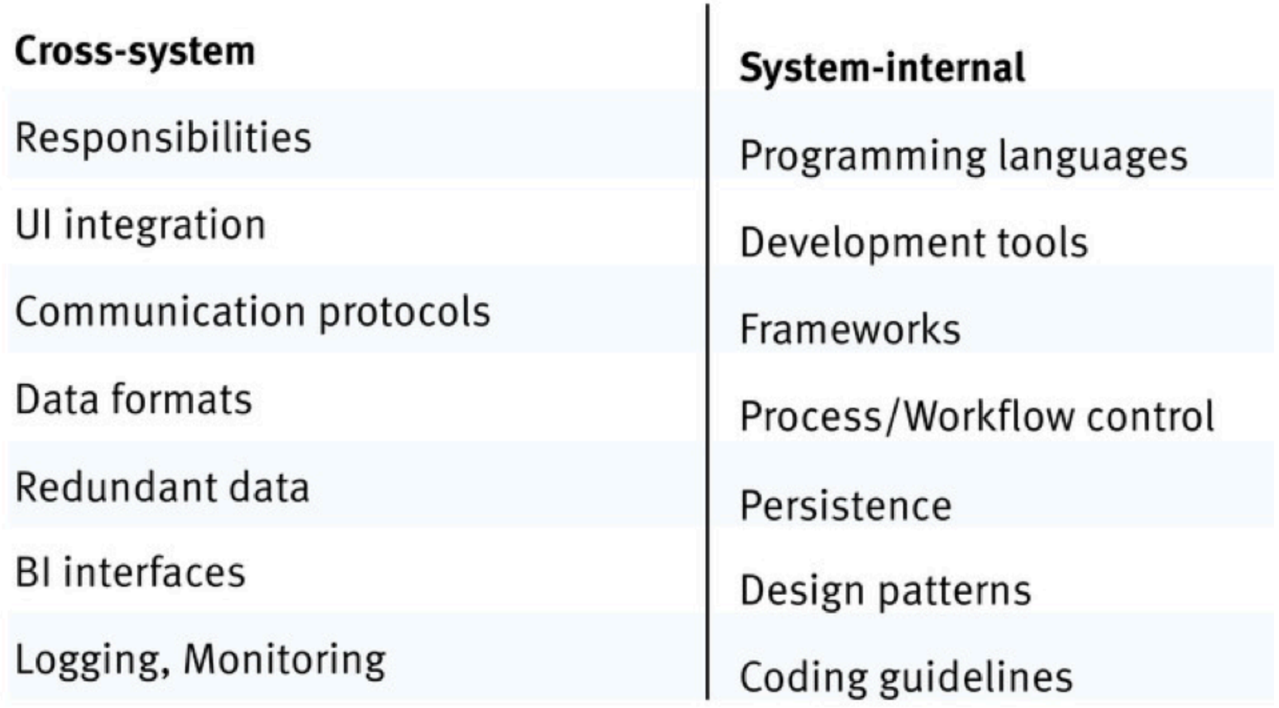
Locomotives – 10 rows

locoId	trainType	departure	destination	obnDnsMismatch	isActive	isDelayed	isDiverted	isTerminatedShort	isAnnulled	isCompleted	isOutOfService	fuel	temperature	oilPressure	speed	RPM	engineLoad
TRN0	PRIVATE	Buckingham, FL	St. Louis, MO	MISMATCH	false	true	true	true	false	false	false	46.0%	113.0	0.0%	0	0	0.0%
TRN1	FREIGHT	New Orleans, LA	Youngstown, OH	MISMATCH	true	true	true	false	false	false	false	43.0%	125.0	0.0%	0	0	0.0%
TRN2	PASSENGER	Manistee, MI	Wilmington, NC	OK	true	false	true	false	false	false	false	12.0%	249.0	3.0%	75	9208	86.32%
TRN3	FREIGHT	Wyndmoor, PA	Waycross, GA	OK	false	false	false	false	false	false	true	75.0%	75.0	0.0%	0	0	0.0%
TRN4	FREIGHT	Princeton, IL	Hampton, VA	MISMATCH	true	true	true	false	false	false	false	80.0%	75.0	0.0%	0	0	0.0%
TRN5	PASSENGER	Chambersburg, PA	Pensacola, FL	MISMATCH	true	false	true	false	false	false	false	68.0%	87.0	17.0%	80	8723	75.6%
TRN6	FREIGHT	Brooklyn, IL	Dunbar, VA	OK	true	false	true	false	false	false	false	12.0%	249.0	20.0%	72	8337	32.51%
TRN7	FREIGHT	Diamonhead, MS	Philadelphia, PA	MISMATCH	false	true	true	true	false	false	false	82.0%	75.0	0.0%	0	0	0.0%
TRN8	FREIGHT	Williamston, NC	Pensacola, FL	OK	false	false	false	false	false	false	true	23.0%	205.0	0.0%	0	0	0.0%
TRN9	PRIVATE	Ludington, MI	Jacksonville, FL	MISMATCH	true	true	true	false	false	false	false	40.0%	137.0	0.0%	0	0	0.0%

Agenda

- Digitalization
- Enterprise Service Bus
- Microservices
- Architecture and Requirements
- **Challenges**

Necessary Rules and Guidelines



Cross-system	System-internal
Responsibilities	Programming languages
UI integration	Development tools
Communication protocols	Frameworks
Data formats	Process/Workflow control
Redundant data	Persistence
BI interfaces	Design patterns
Logging, Monitoring	Coding guidelines

Stefan Tilkov, https://speakerdeck.com/stilkov/microservices-talk-berlin

Avoid a zoo of technologies and frameworks!

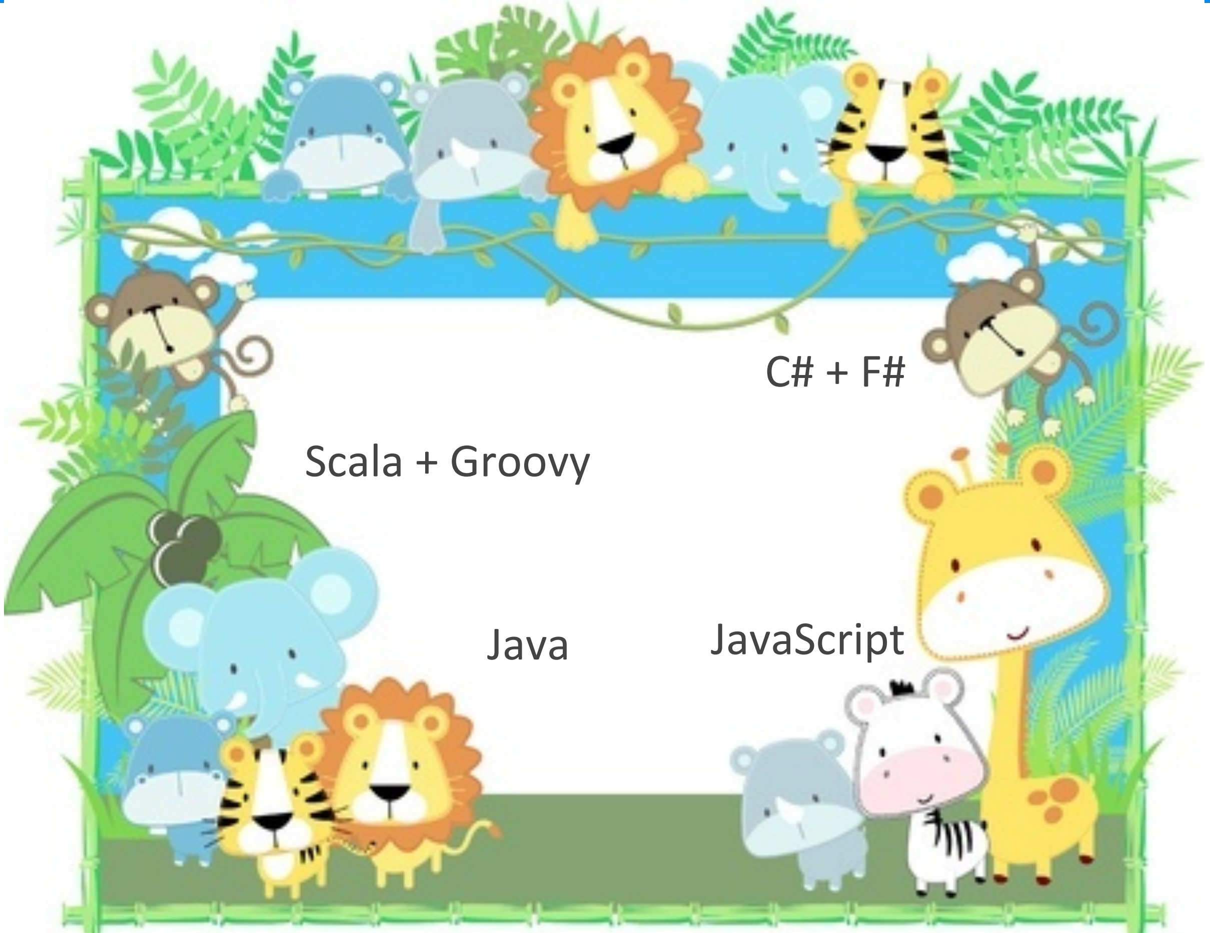
Java EE
App Server

Java Process
(JAR File)

Node.js
Server

.NET
Platform

Middleware
Platform



C# + F#

Scala + Groovy

Java

JavaScript

Framework 1

Framework 2

Framework X

No Framework

Tool 1

Tool 2

Tool X

No Tool

DevOps / Continuous Delivery

Theory

Every team is responsible for development, test, deployment and operations.

Therefore, technology and tool choice do not matter.

Practice

People and intellectual property leave companies.

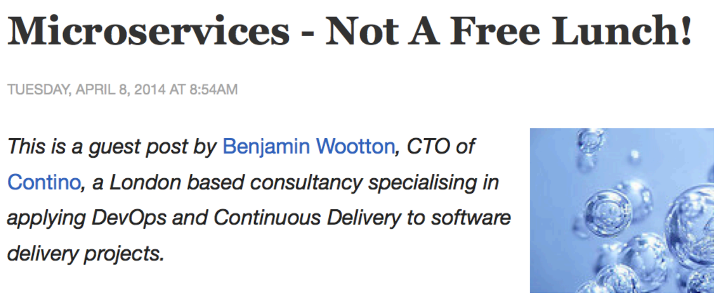
An enterprise strategy exists to reduce risks and costs.

The team cannot control everything (e.g. when using Open API or SaaS services).

TIBCO™

Microservices is a lot of effort!

- Significant operations overhead
- Substantial DevOps skills required
- Implicit interfaces
- Duplication of effort
- Distributed system complexity
- Asynchronicity is difficult
- Testability Challenges



Microservices - Not A Free Lunch!

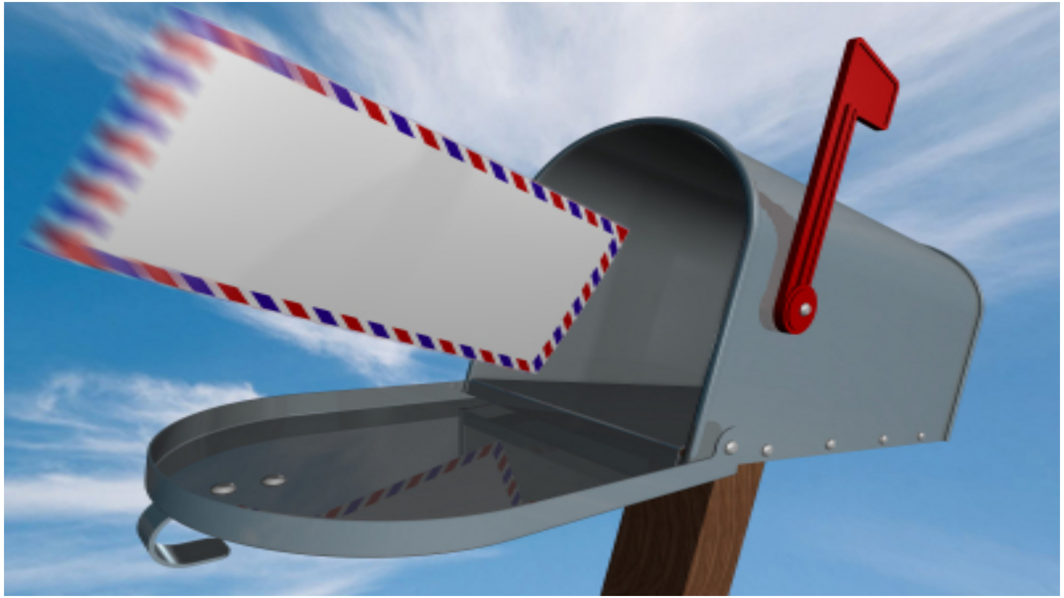
TUESDAY, APRIL 8, 2014 AT 8:54AM

This is a guest post by Benjamin Wootton, CTO of Contino, a London based consultancy specialising in applying DevOps and Continuous Delivery to software delivery projects.

http://highscalability.com/blog/2015/4/8/microservices-not-a-free-lunch.html

”[…] when considering Microservice like architectures, it's really important to not be attracted to the hype on this one as the challenges and costs are as real as the benefits."

Did you get the Key Message?



Key Messages

- Microservices = SOA done right! ☑
- Integration is key for success – the product name does not matter! ☑
- Real time event correlation is the game changer! ☑

Questions?

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