

Governance, Registries and Monitoring

Oxford University

Software Engineering Programme

Dec 2014



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).

Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.

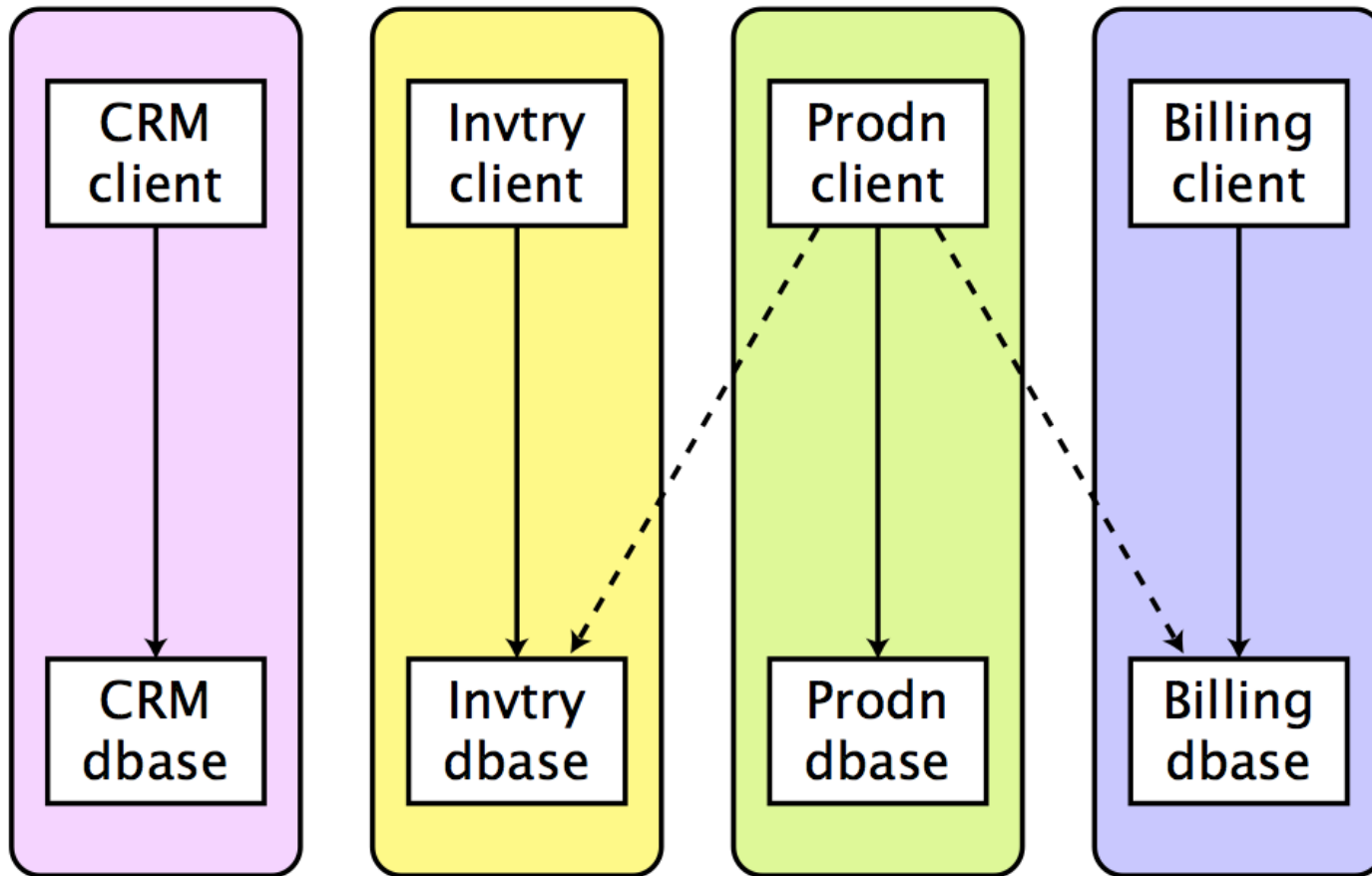
See <http://creativecommons.org/licenses/by-sa/3.0/>

Contents

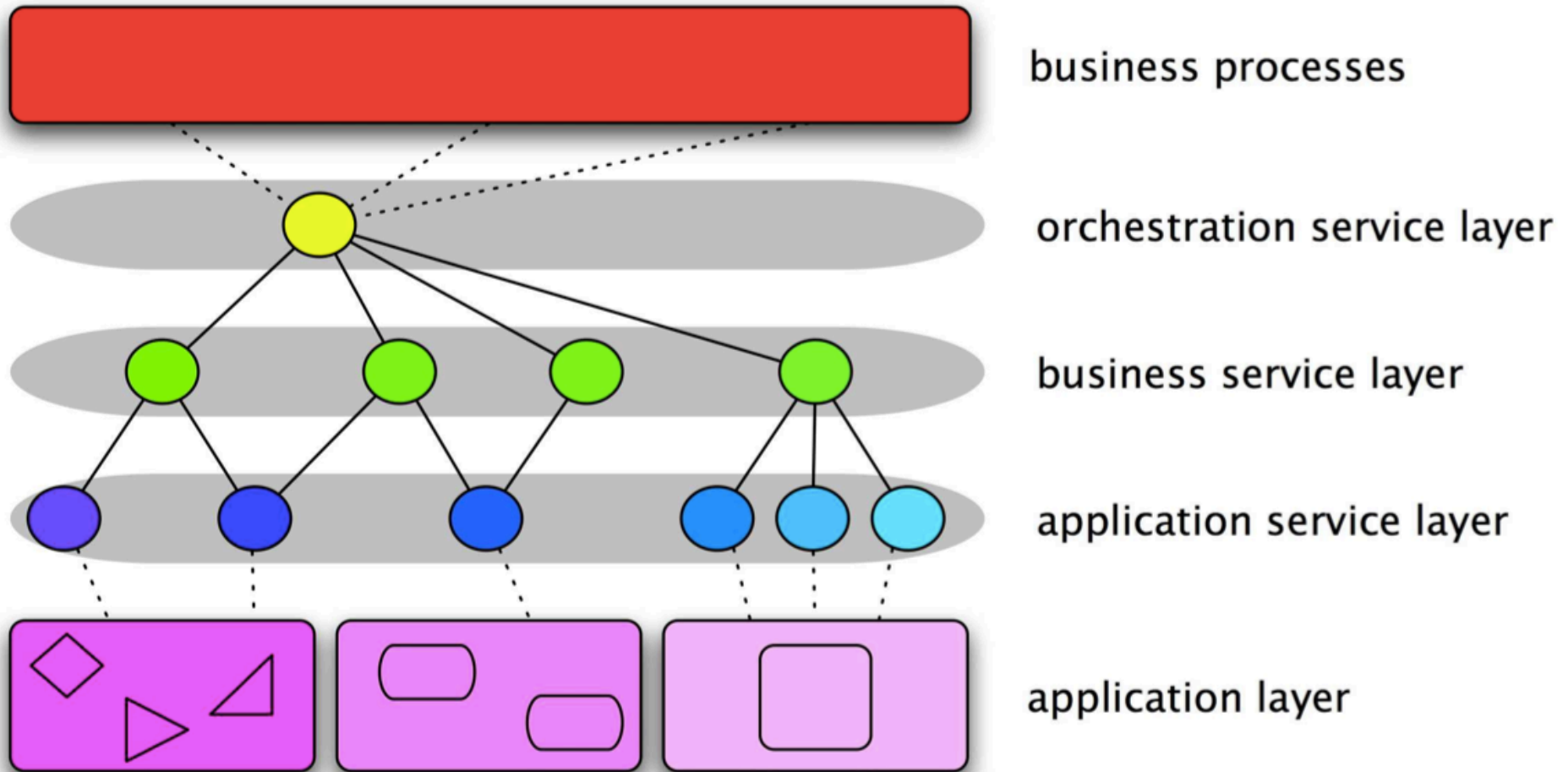
- Software Development Lifecycle
- Registries
- Design Governance
- Runtime Governance



Before SOA



With SOA



SOA has an impact on organization

- Refactoring of fiefdoms:
 - backend departments
 - cross-domain departments – frontend departments
 - “solutions managers”
- Requires collaboration and trust
- May change the funding model
 - That will pull in resistance



Conway's Law

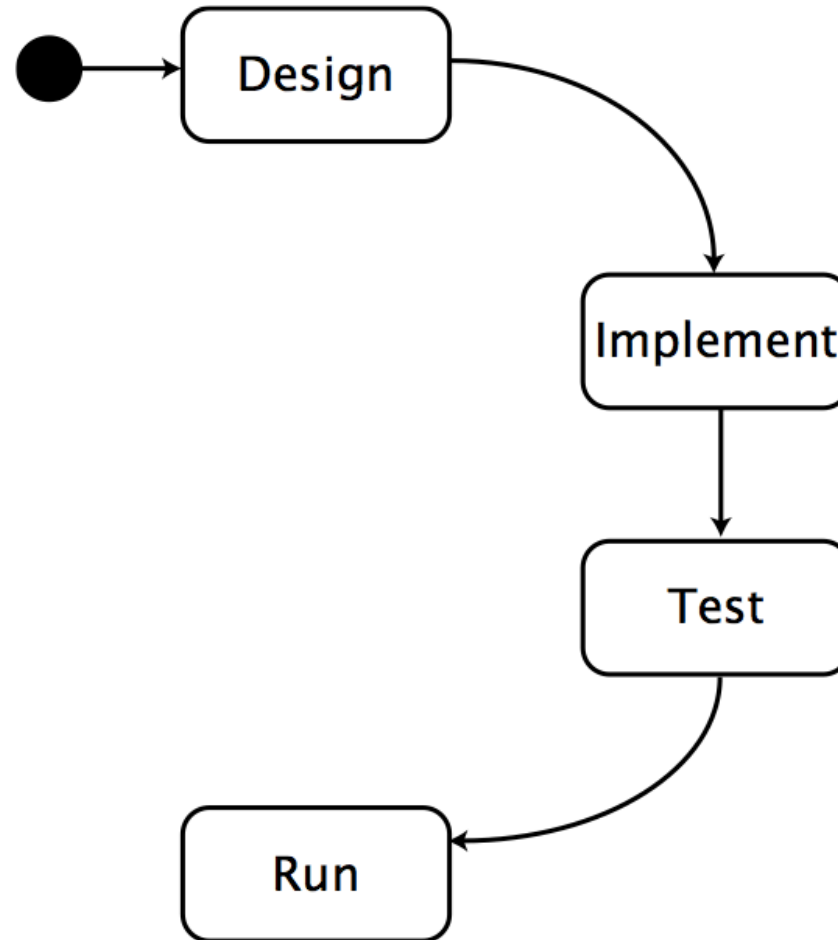
- Any organization that designs a system will inevitably produce a design whose structure is a copy of the organization's communication structure.

Melvin Conway, *How Do Committees Invent?*,
Datamation Apr 1968,
<http://www.melconway.com/law/>

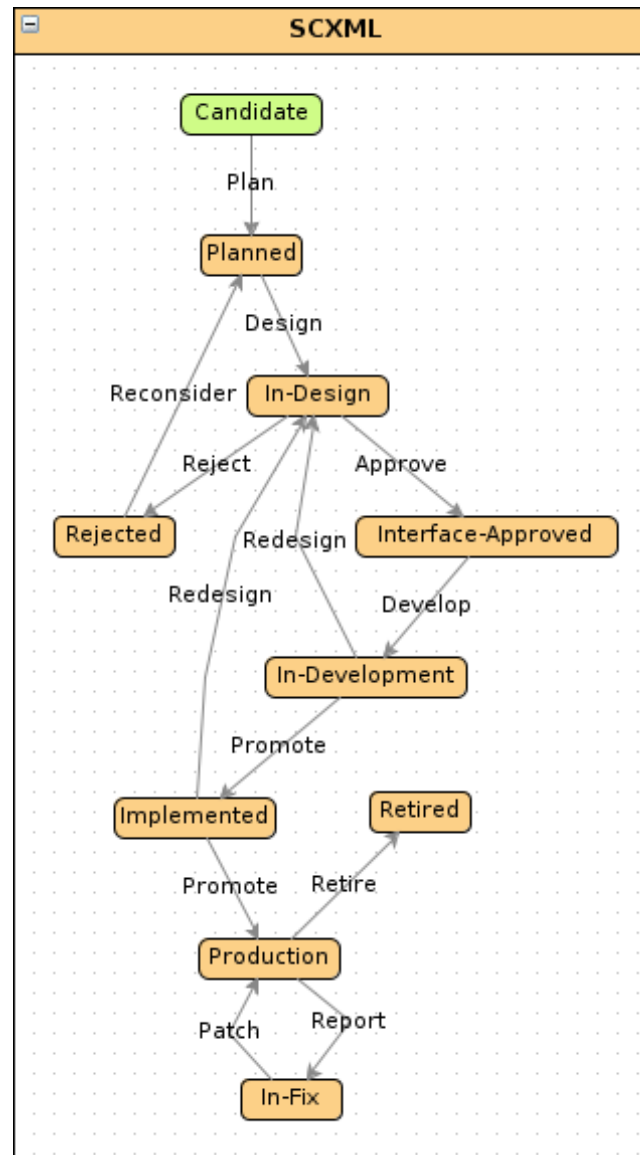
- Popularized and named by Fred Brooks in *The Mythical Man-Month*: “If you have four groups working on a compiler, you’ll get a 4-pass compiler.”



Software Development Lifecycle



Not that simple!



High level governance

- *Visions, objectives, business case, funding model*
 - Why are we doing this? How will we pay for it?
 - *Reference architecture*
Fundamental decisions: preferred technology, message exchange patterns, metamodel, etc
- *Rules and responsibilities*
 - who drives and cares about issues
- *Policies, standards, formats, processes, lifecycles*
 - decide and document, in standard notations



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).

Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.

See <http://creativecommons.org/licenses/by-sa/3.0/>

Technical Governance

- *Documentation*
 - important for transparency; promotes non-technical issues
- *Service management*
 - repositories and registries for services and contracts
- *Monitoring*
 - conformance to policies, meeting SLAs, preparing for withdrawal
- *Change and configuration management*
 - Code lifecycle, DevOps, SOA, the intersection



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).

Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.

See <http://creativecommons.org/licenses/by-sa/3.0/>

Establishing SOA

- *Developer-driven, grass-roots*
 - leads to technological experience; likely to be uncoordinated
- *Business-driven*
 - proof of concept helps adoption; limited benefit from early projects
- *IT-driven*
 - effective for infrastructure; focus on technical aspects
- *Management-driven*
 - top-down coordinated, driven by business priorities; expensive, disruptive, risky



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).

Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.

See <http://creativecommons.org/licenses/by-sa/3.0/>

Design Time Governance



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See <http://creativecommons.org/licenses/by-sa/3.0/>

UDDI

- An XML-ization of the Windows Registry
 - Lots of nasty UUIDs
- But aimed at Internet Scale
- Launched with the concept of major public UDDI servers (like DNS)
 - Universal Business Registry (UBR)



UDDI

- A UDDI business registration consists of three components:
 - White Pages — address, contact, and known identifiers;
 - Yellow Pages — industrial categorizations based on standard taxonomies;
 - Green Pages — technical information about services exposed by the business.



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).

Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.

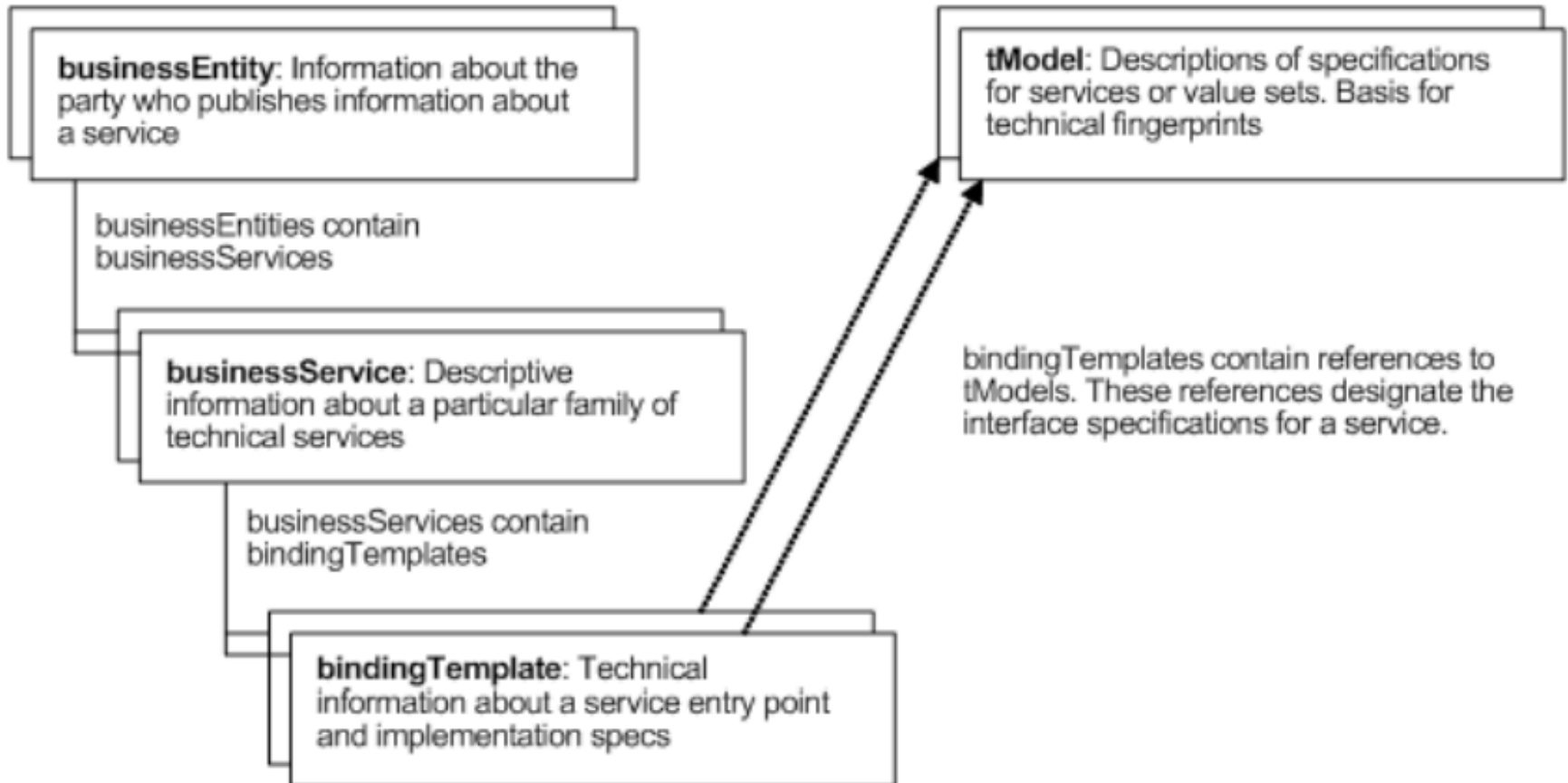
See <http://creativecommons.org/licenses/by-sa/3.0/>

History

UDDI VERSION	YEAR RELEASED	KEY OBJECTIVE
1.0	2000	Create foundation for registry of Internet-based business services
2.0	2001	Align specification with emerging Web services standards and provide flexible service taxonomy. Formally released under OASIS aegis in 2003
3.0	2004	Support secure interaction of private and public implementations as major element of service-oriented infrastructure. To be released by OASIS in late 2004



Key Data Types



UDDI problems

- tModel concept too complex and unwieldy
 - Lack of any standard tModels
- No *simple* link into WSDL
- Doesn't address the major issues of Service Registry and Governance
 - Lifecycle management
 - Notification
 - Dependency Management



SOA Repository Artefact Model and Protocol (S-RAMP)

- In progress at OASIS
- Based on Atom/AtomPub concepts
- Aimed at solving the real world problems left unanswered by UDDI
 - Which WSDLs and schemas import a particular schema
 - Which of my services are in production?
 - Which services are governed by a specific SLA?
 - What is the latest version of service X?



Aspects of “real” registries

- Meta-Modelling
- Taxonomies
- Versioning
- Associations and Properties
- Lifecycle Management
- Dependency Management
- Repository
- Search
- Machine and Human interfaces



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).

Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.

See <http://creativecommons.org/licenses/by-sa/3.0/>

Meta-Modelling

- Creating / modifying the model to support new artefacts
 - e.g. Teams, Projects, Organizations, etc
- Also used for extending more technical attributes
 - e.g. adding WADL or Swagger support



Versioning

- Micro-versioning/revision management
 - Keeping track of every minor update to a WSDL
 - Permanent URLs for given versions
- “Business” Versioning
 - Service A is available as
 - 1.2.3 deprecated
 - 2.5.1 current



Associations and Properties

- Properties
 - General name / value pairs attached to resources
- Associations
 - Named Links between resources
 - e.g.
 - A isUsedBy B
 - B isManagedBy C



Lifecycle Management

- Each service in the corporate datacentre **MUST**:
 - Start as “In Design”
 - Be approved by the Design Review Team
 - Iterate through Development
 - Pass validation tests before entering Staging
 - Be approved by the Security and Performance Teams before entering Production
 - Be deprecated when no longer supported



Dependency Management

- Each Service S_n depends on Schemas $\{Y_{1..n}\}$
- Schema Y depends on Schemas $\{Z_{1..n}\}$
- Schemas are shared between services
- Owners and users of services need to be made aware of new versions of schemas they depend on (even if they didn't know it!)



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).

Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.

See <http://creativecommons.org/licenses/by-sa/3.0/>

Interfaces

- Registries are used by humans, but shouldn't always be!
- e.g. Maven build rather than forcing developers to use a website
 - One company I know hires a “Registry Monkey” who ONLY enters services into a registry
 - Each service takes 83 steps
 - He hasn't yet committed suicide



Registry, DevOps, SCM

- Ideally need to connect:
 - The Source Code Management (CVS, SVN, Git)
 - The build and test environment
 - Hudson, Jenkins, Bamboo
 - Selenium, JUnit, etc
 - The production management process
 - DevOps, Puppet, Chef
 - The design time registry
 - The runtime registry



Runtime Governance

- Finding services at runtime
- Monitoring services at runtime
- Managing SLAs
- Correlation
- Acting on situations



WS-Discovery

- A simple(r) protocol just aimed at finding “live” endpoints
- For example used by HP printers for network printing
 - Though my own experience is that mine is a bit hit and miss!
- Supports UDP and TCP
 - Managed or Ad-Hoc mode



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).

Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.

See <http://creativecommons.org/licenses/by-sa/3.0/>

WS-Discovery “ad-hoc”

Screen Clipping



WS-Discovery Message Types

- Hello messages
- Bye messages
- Probe messages
- Probe match messages
- Resolve messages
- Resolve match messages



Hello

```
<wsd:Hello>
  <wsa:EndpointReference >
    <wsa:Address>urn:uuid:50bccd63-8c1f-4f9d-a0db
    </wsa:Address>
  </wsa:EndpointReference>
  <wsd:Types xmlns:a="http://echo..wso2.org">a:echoPortType
  </wsd:Types>
  <wsd:Scopes>http://.../ws-dd/ns/discovery/2009/01/DefaultScope
  </wsd:Scopes>
  <wsd:XAddrs>https://localhost:8243/services/echo
    http://localhost:8280/services/echo</wsd:XAddrs>
  <wsd:MetadataVersion>1</wsd:MetadataVersion>
</wsd:Hello>
```



Runtime Monitoring



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See <http://creativecommons.org/licenses/by-sa/3.0/>

Aggregation

- Gathering data
- How to collect data efficiently
- How to store data effectively
- What data to capture

3) How long have you been coming to the Fair? (Circle choice)
Less than 1 year 1-2 years 3-5 years 6-10 years 12+ years

2) Please rate your experience at the Fair:

	Excellent	Good	Poor	Unacceptable
Overall Experience	✓			
Venue	✓			
Vendors (variety & quality)	✓			
Speakers	✓			

3) What would you like to see at the Fair? (Circle choices)
Knitting Spinning
Accessories (buttons, beads, bags, etc.) Crocheting
Other:

Please use the back of this page to respond to the following:

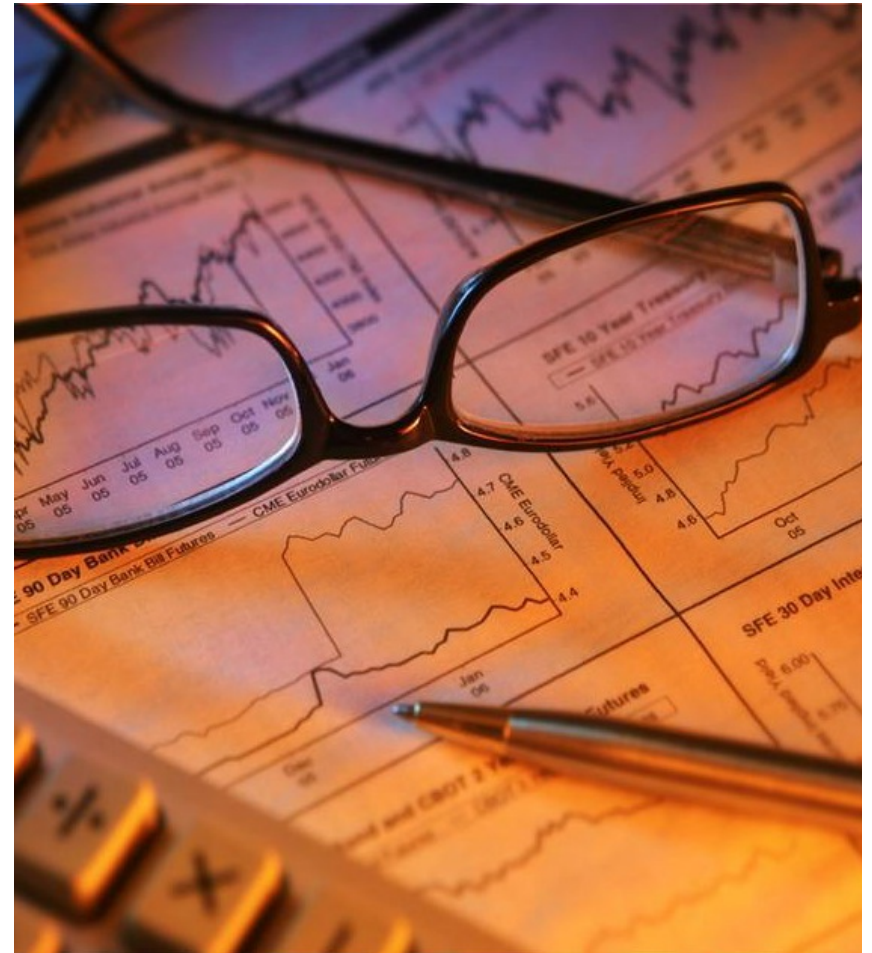
4) What are your favourite things about the Fair? YARN!

5) What would you improve about the Fair? MORE!



Analysis

- Data operations
- Defining KPIs and analytics
- Operating on large amounts of historical or current data
- Creating intelligence



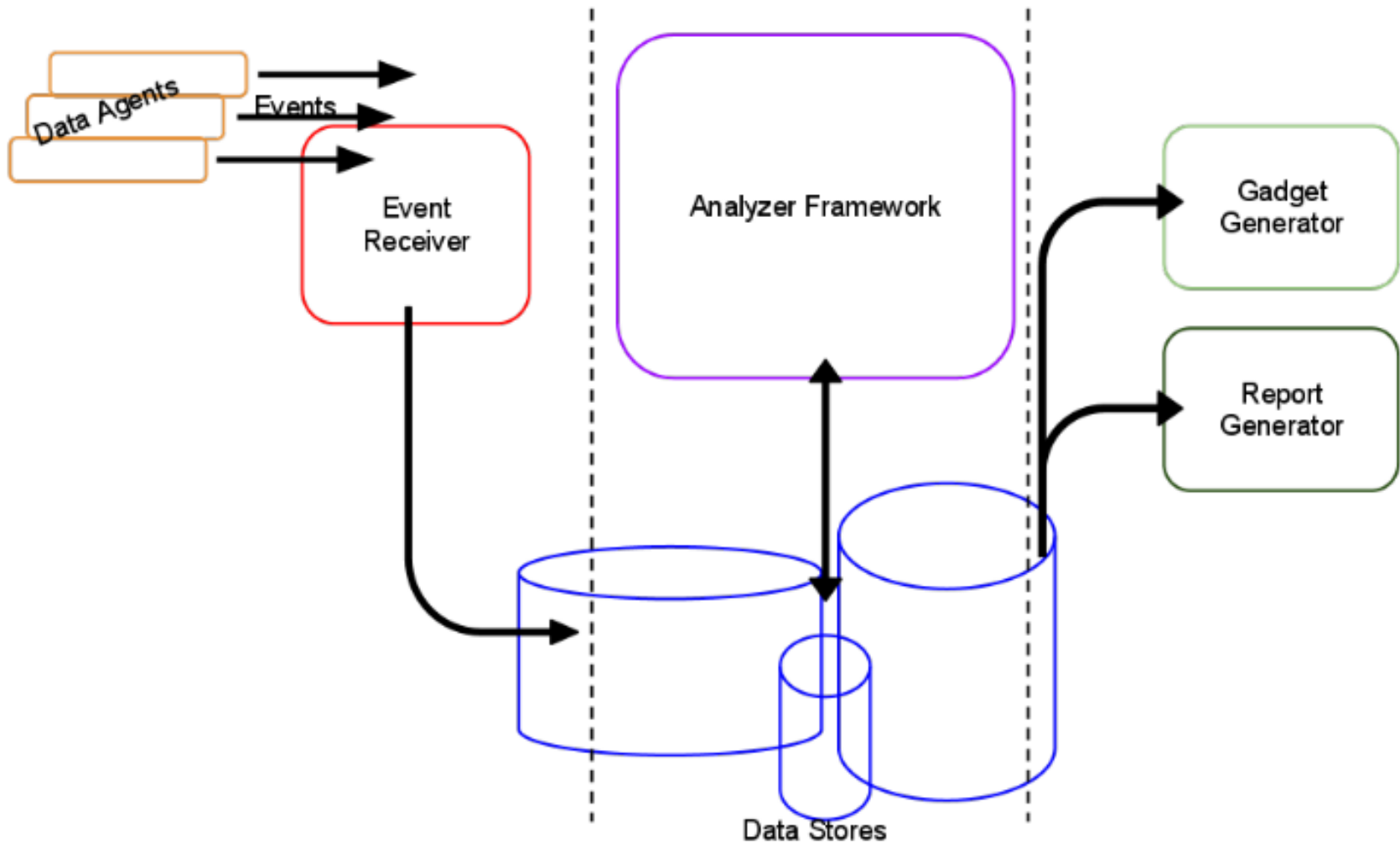
Presentation

- Visualization
- Dashboards
- Reports



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).
Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See <http://creativecommons.org/licenses/by-sa/3.0/>

BAM visually

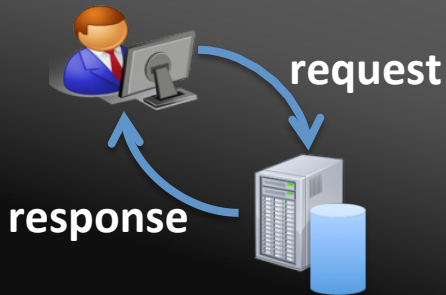
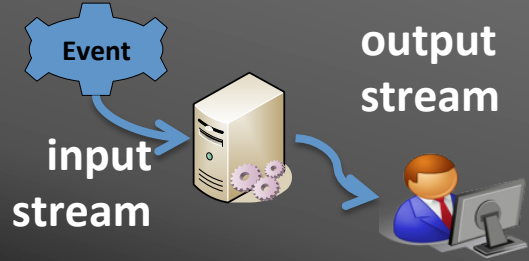


Closing the loop

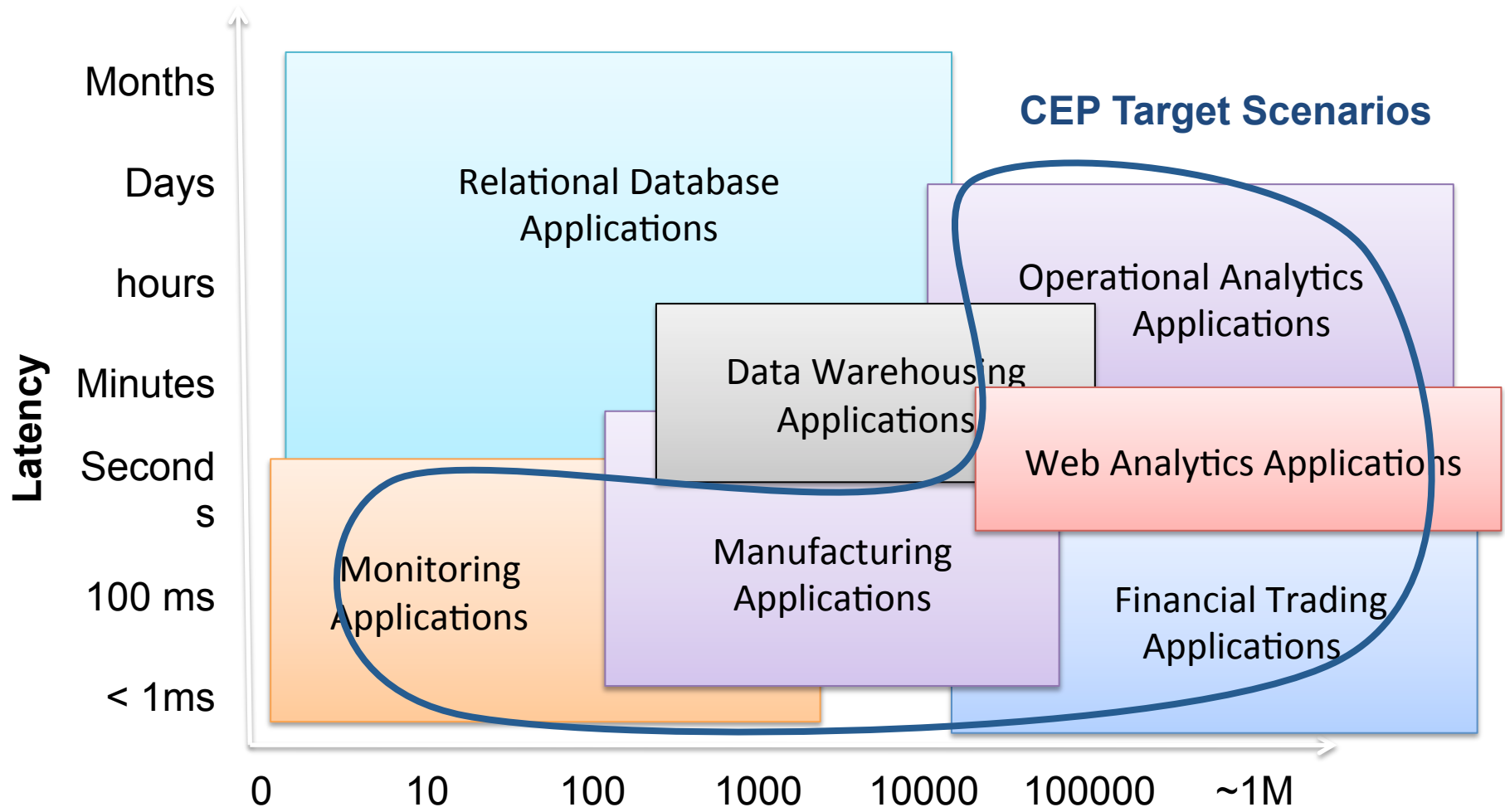
- SLAs are time based rules about performance data
 - Is service X responding in under 50ms for more than 99% of calls within the last 5 mins?
 - Does the sales team respond to leads within 4 hours?
 - Has the average CPU utilization over the last day gone more than 50% higher than the weekly average



Event Processing ?

	Storage based Applications	Event-driven Applications
Query Paradigm	Ad-hoc queries or requests	Continuous standing queries
Latency	Seconds, hours, days	Milliseconds or less
Data Rate	Hundreds of events/sec	Tens of thousands of events/sec or more
		

Scenarios of Event Processing



Event Processing

- Simple Event Processing
 - Simple filters (e.g. Is this a gold or platinum customer?)
- Event Stream Processing
 - Looking across multiple event streams and joining multiple event stream etc.
- Complex Event Processing
 - Processing multiple event streams to identify meaningful patterns, using complex conditions & temporal windows
 - E.g. There has been a more than 10% increase in overall trading activity AND the average price of commodities has fallen 2% in the last 4 hours



© Paul Fremantle 2012. Portions © Jeremy Gibbons 2010, © WSO2 2005-2012 used with permission of the author(s).

Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.

See <http://creativecommons.org/licenses/by-sa/3.0/>

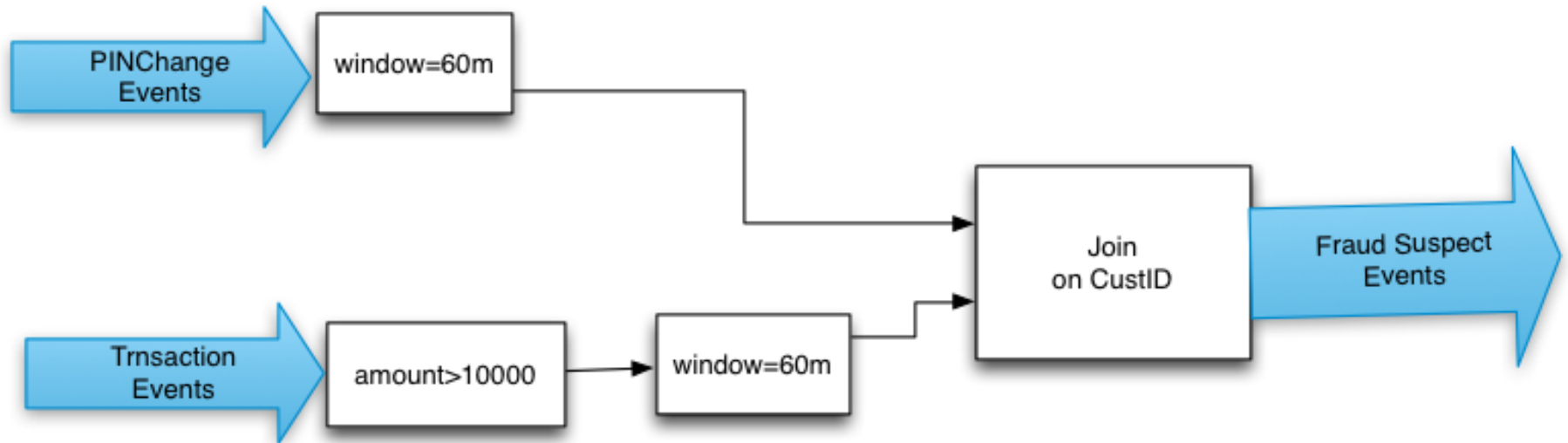
CEP Queries

- Types of queries are following
 - Filters and Projection
 - Windows – events are processed within temporal windows (e.g. for aggregation and joins).
Time window vs. length window.
 - Ordering – identify event sequences and patterns (e.g. for a credit card new location followed by small and a large purchase might suggest a fraud)
 - Joins – join two streams



Example Query

```
from p=PINChangeEvents#window.time(3600) join  
    t=TransactionEvents[amount>10000]#window.time(3600)  
    on p.custid==t.custid  
return t.custid, t.amount;
```



Acting on runtime situations

- SOA infrastructure →
 - Business Activity Monitoring stream →
 - Complex Event Processing rules →
 - SOA infrastructure
-
- e.g. when utilization of my cloud web farm is $> 75\%$ start a new server



Resources

- UDDI
 - <http://uddi.xml.org/>
- S-RAMP
 - https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=s-ramp
- WS-Dynamic Discovery
 - <http://docs.oasis-open.org/ws-dd/ns/discovery/2009/01>
- WSO2 Governance Registry
 - <http://wso2.com/products/governance-registry/>

