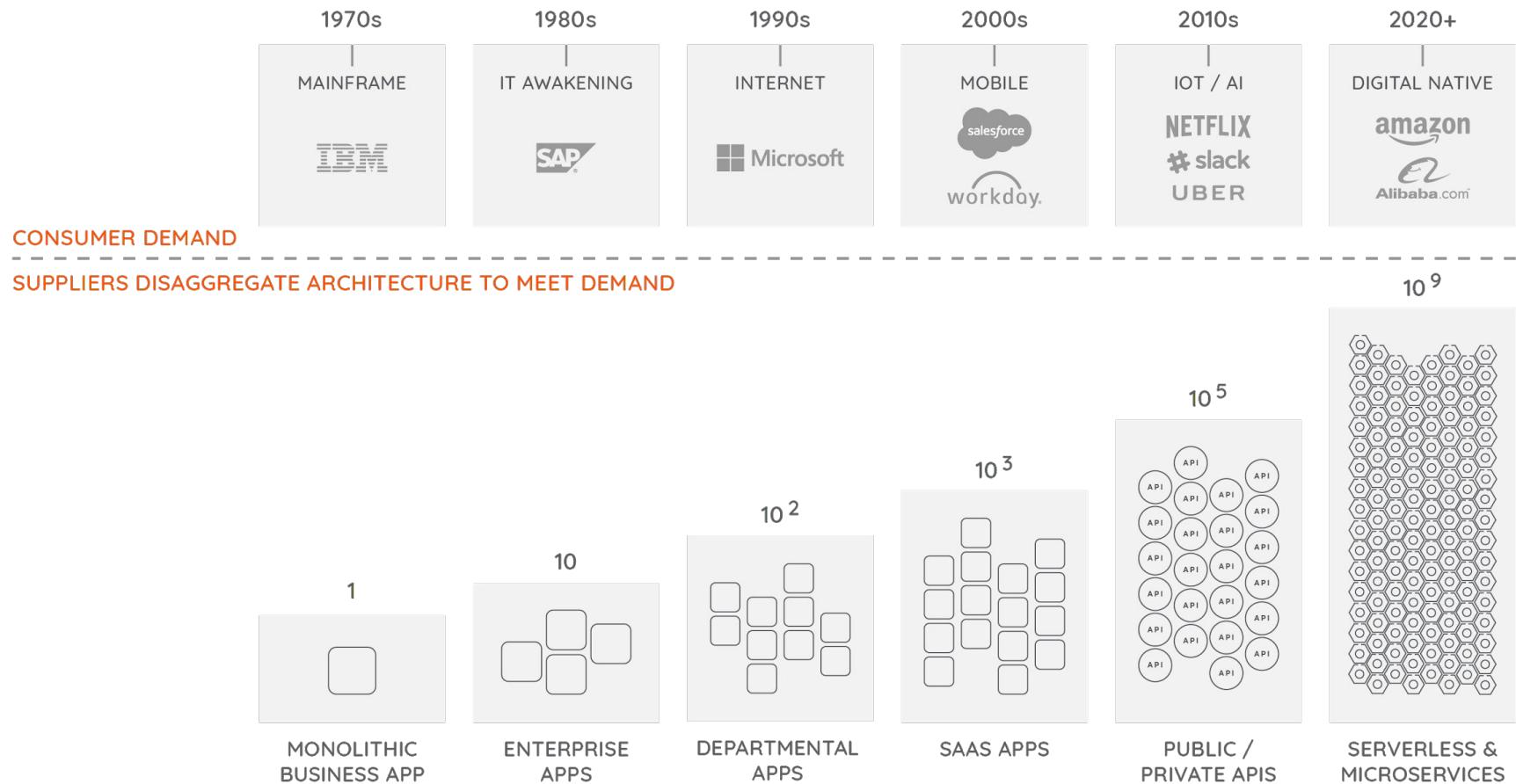


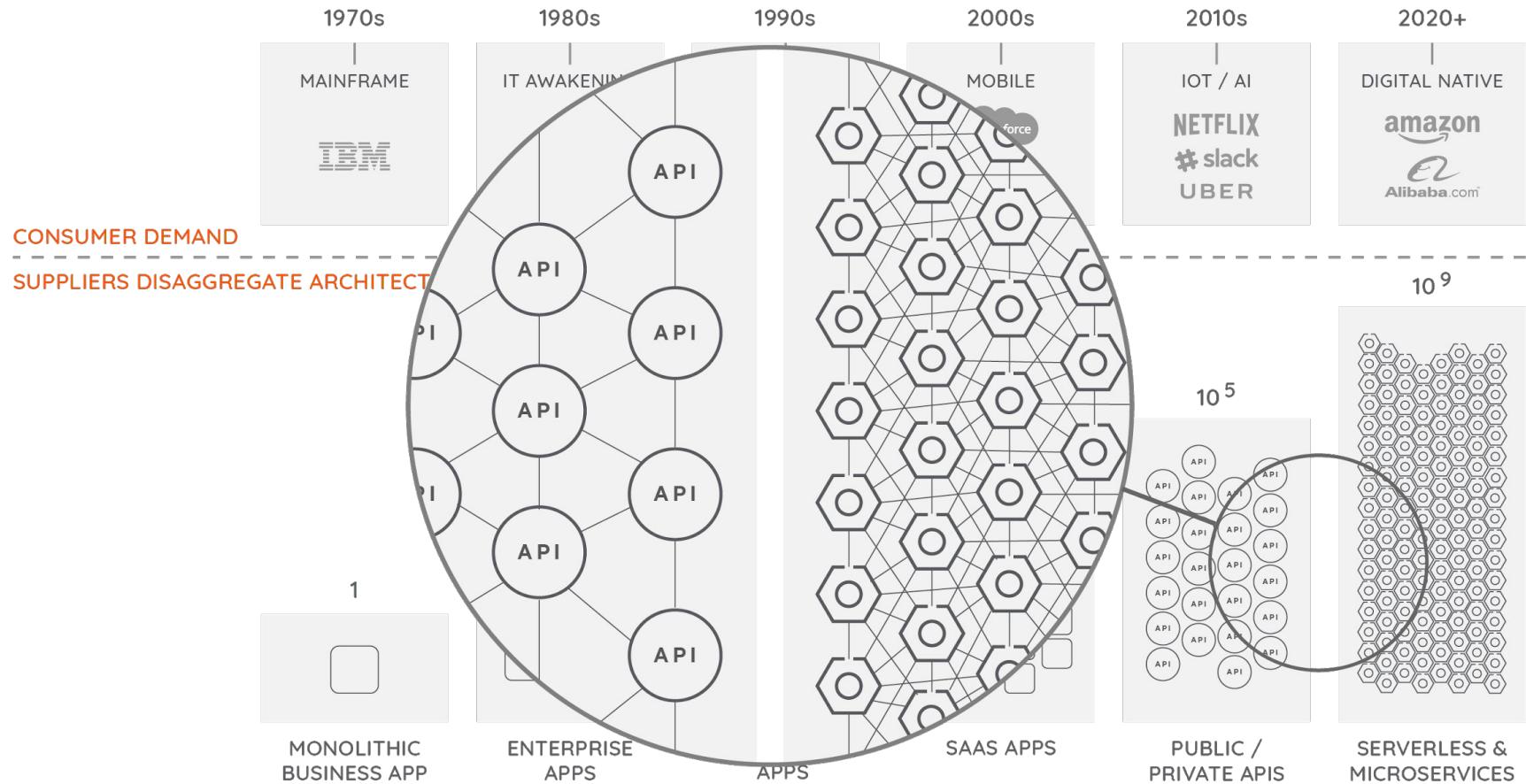
Motivation

Oxford University
Software Engineering Programme
April 2021

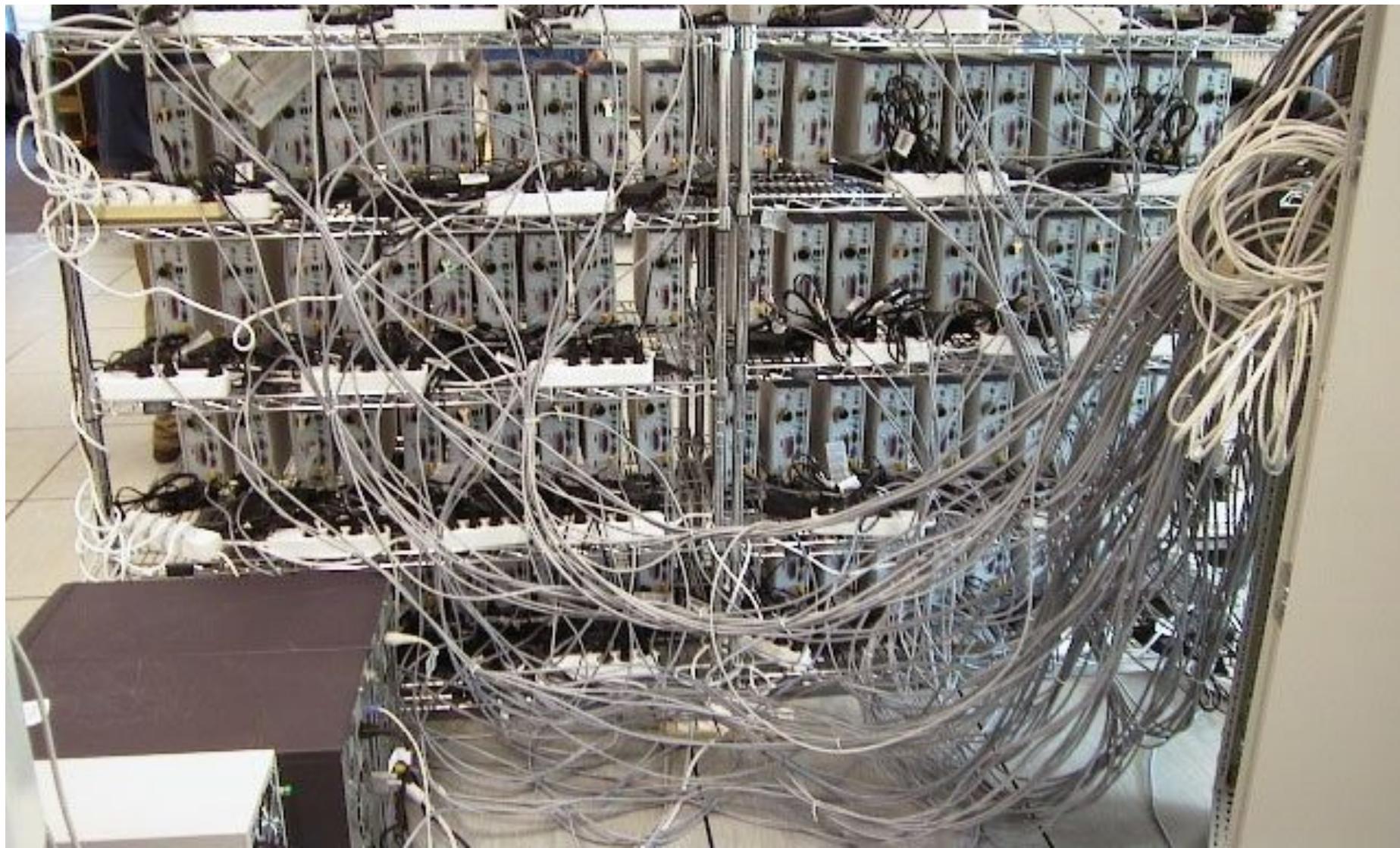


© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>





Distributed Computing



Why is Distributed Computing important?

- Scale
- Evolution
- Integration
- Cross-boundary
- Resilience



© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Learning from

THE AMAZON

TECHNOLOGY

PLATFORM

Many think of Amazon as “that hugely successful online bookstore.” You would expect Amazon CTO Werner Vogels to embrace this distinction, but in fact it causes him some concern. “I think it’s important to realize that first and foremost Amazon is a technology company,” says Vogels. And he’s right. Over the past years, Vogels has helped Amazon grow from an online retailer (albeit one of the largest, with more than 55 million active customer accounts) into a platform on which more than 1 million active retail partners worldwide do business. Behind Amazon’s successful evolution from retailer to technology platform is its SOA (service-oriented architecture), which broke new technological ground and proved that SOAs can deliver on their promises.

Vogels came to Amazon from Cornell University, where he was working on high-availability systems and the management of scalable enterprise systems. He maintains that research spirit at Amazon, which regularly must solve problems never before encountered. “Maybe other companies call it research. We just call it development,” he points out.

Interviewing Vogels is ACM Turing Award winner and Microsoft Technical Fellow Jim Gray.

application, running on a Web server, talking to a database on the back end. This application, dubbed Obidos, evolved to hold

all the business logic, all the display logic, and all the functionality that Amazon eventually became famous for: similarities, recommendations, Listmania, reviews, etc.

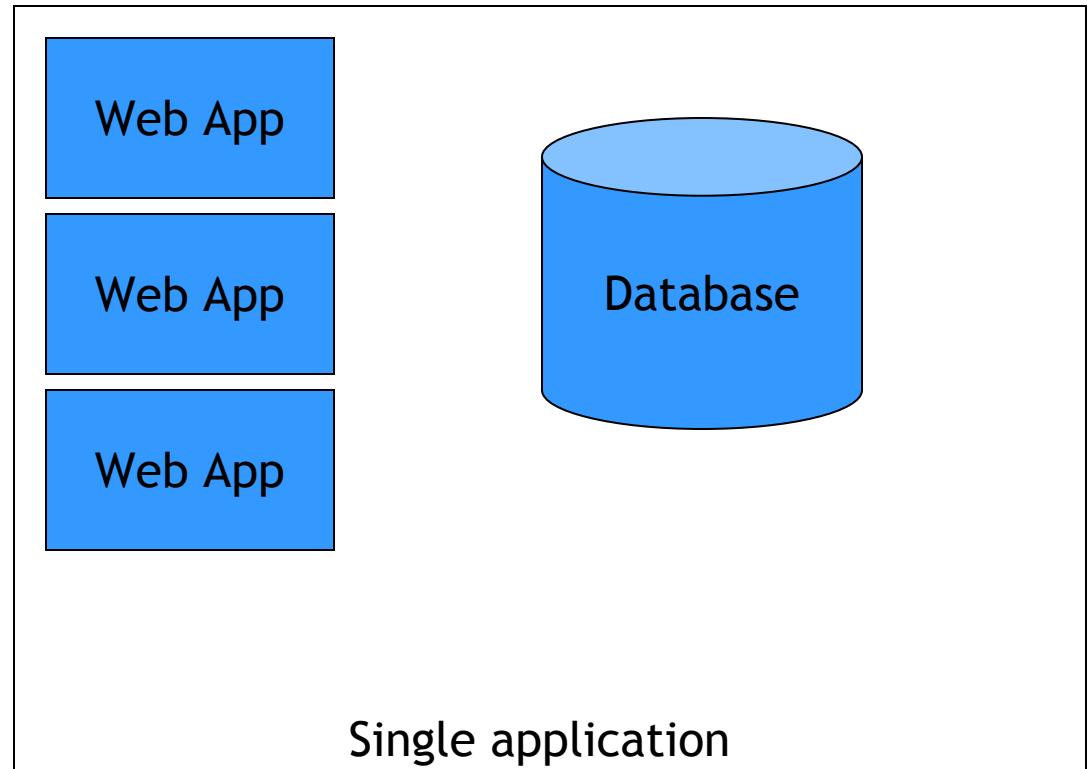
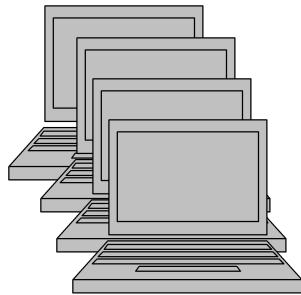


Source: Interview with Werner Vogels, ACM Queue
<https://queue.acm.org/detail.cfm?id=1142065>



© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>

“Obidos”

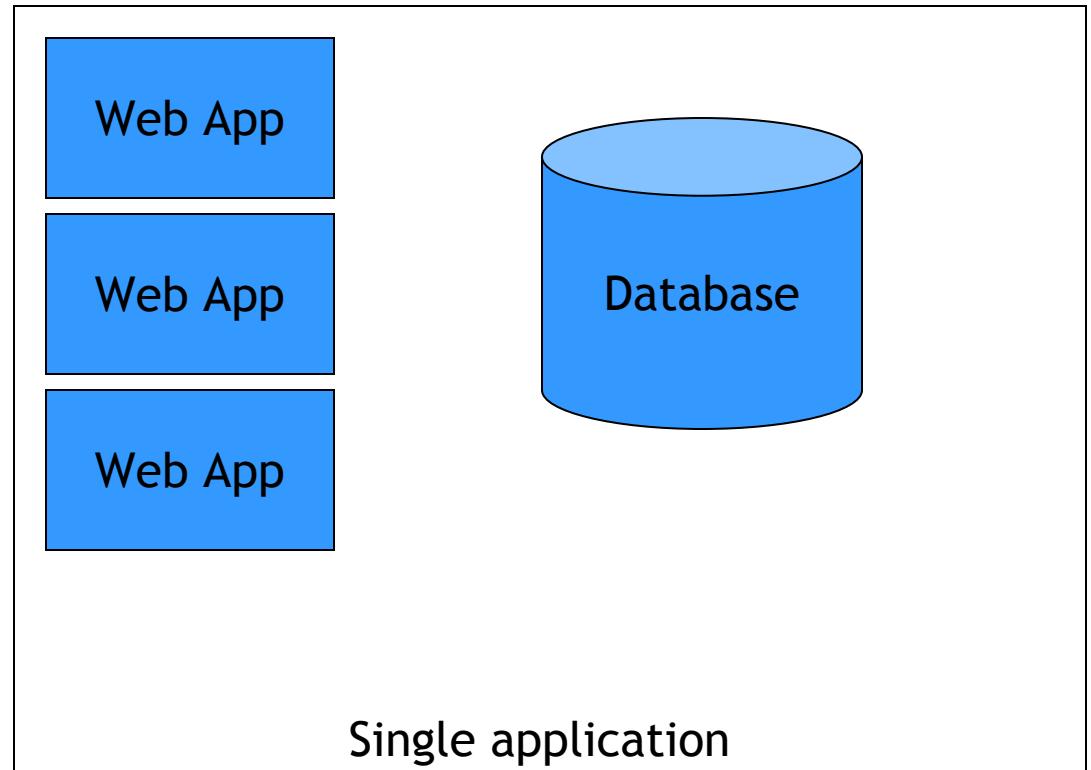
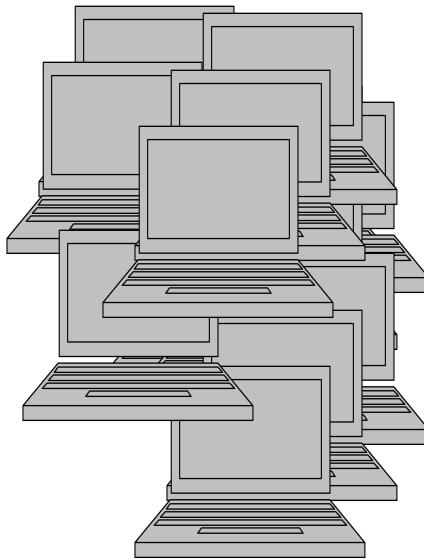


But it was Successful!

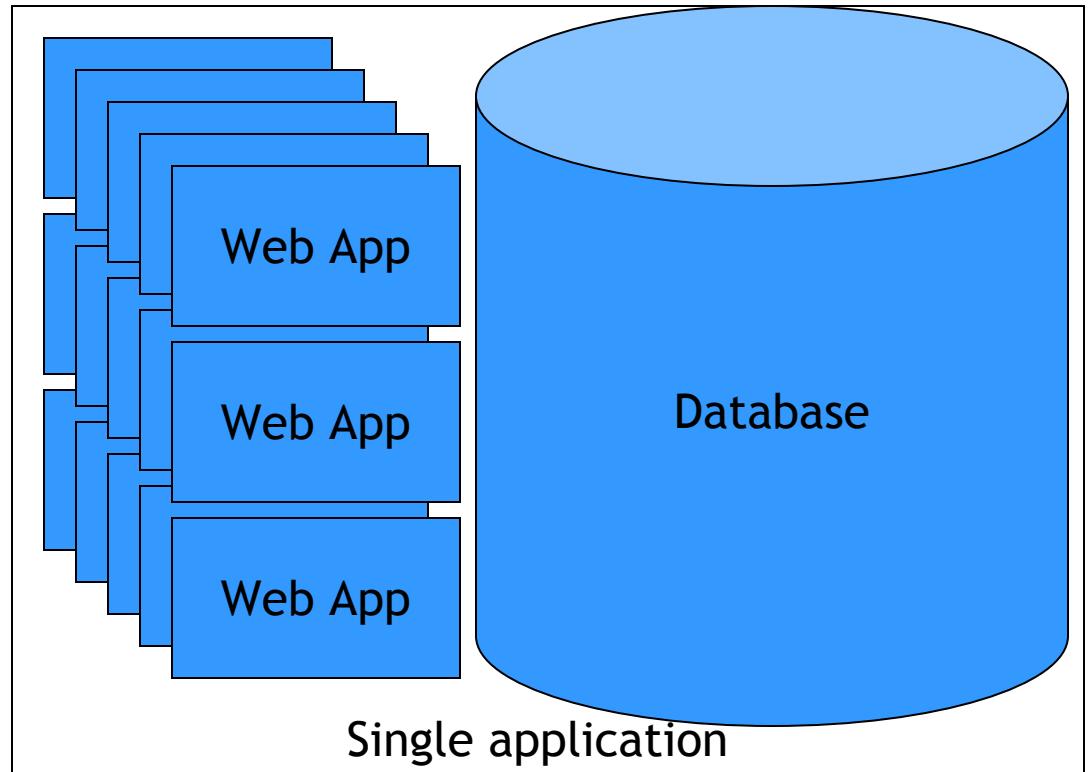
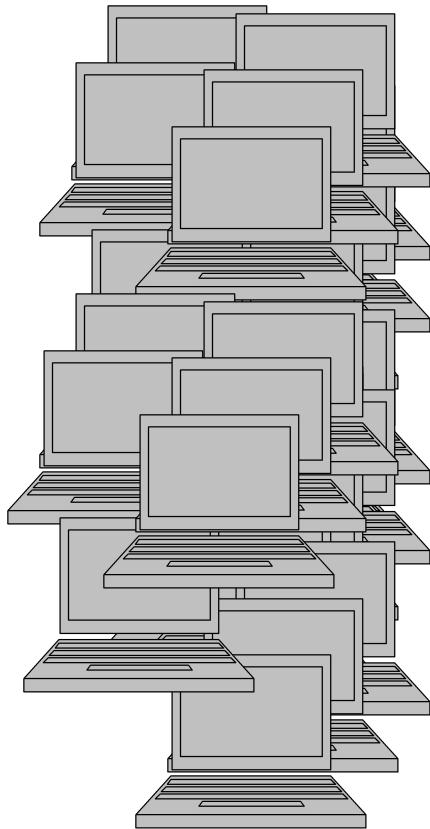


© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Internet Scale Up



... to bursting point



Problems

- Too many complex pieces of software in a single system
- No evolution possible
- Need to scale independently
 - Parts sharing resources with other unknown code paths
- No isolation
- No clear **ownership**



Database scaling

- Databases a shared resource
- Hard to scale-out
- Front-end and backend shared by
 - Too many teams
 - Too many processes



© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>

A new model

- In 2001/2 decided on a new approach
- SOA based – even before the term was in common usage
- Encapsulating the data with the business logic that operates on the data
- Only access through a published service interface
- No direct database access is allowed from outside the service
- No data sharing among the services.



So one day Jeff Bezos issued a mandate. He's doing that all the time, of course, and people scramble like ants being pounded with a rubber mallet whenever it happens. But on one occasion -- back around 2002 I think, plus or minus a year -- he issued a mandate that was so out there, so huge and eye-bulgingly ponderous, that it made all of his other mandates look like unsolicited peer bonuses.

His Big Mandate went something along these lines:

1. All teams will henceforth expose their data and functionality through service interfaces.
2. Teams must communicate with each other through these interfaces.
3. There will be no other form of interprocess communication allowed: no direct linking, no direct reads of another team's data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.
4. It doesn't matter what technology they use. HTTP, Corba, Pubsub, custom protocols -- doesn't matter. Bezos doesn't care.
5. All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.
6. Anyone who doesn't do this will be fired.
7. Thank you; have a nice day!

Ha, ha! You 150-odd ex-Amazon folks here will of course realize immediately that #7 was a little joke I threw in, because Bezos most definitely does not give a shit about your day.



Growth

- Amazon services in the hundreds
- A typical visit to the homepage may include calls to 100 services
- Caching reduces the actual network traffic
- Fully distributed, decentralized
- The web servers are just one client into the service fabric



Matched by business growth

- Amazon is supporting many new businesses
- Books, CDs, Electronics, Toys, Tools and Hardware,...
- Plus millions of independent retailers sharing the Amazon platform



Lessons learnt

- Isolation
 - Service Orientation promotes ownership and control
- Scalability
 - By preventing direct database access, can scale the services without affecting clients
- Need a common service-access mechanism
 - Aggregation
 - Routing
 - Tracking



Organization

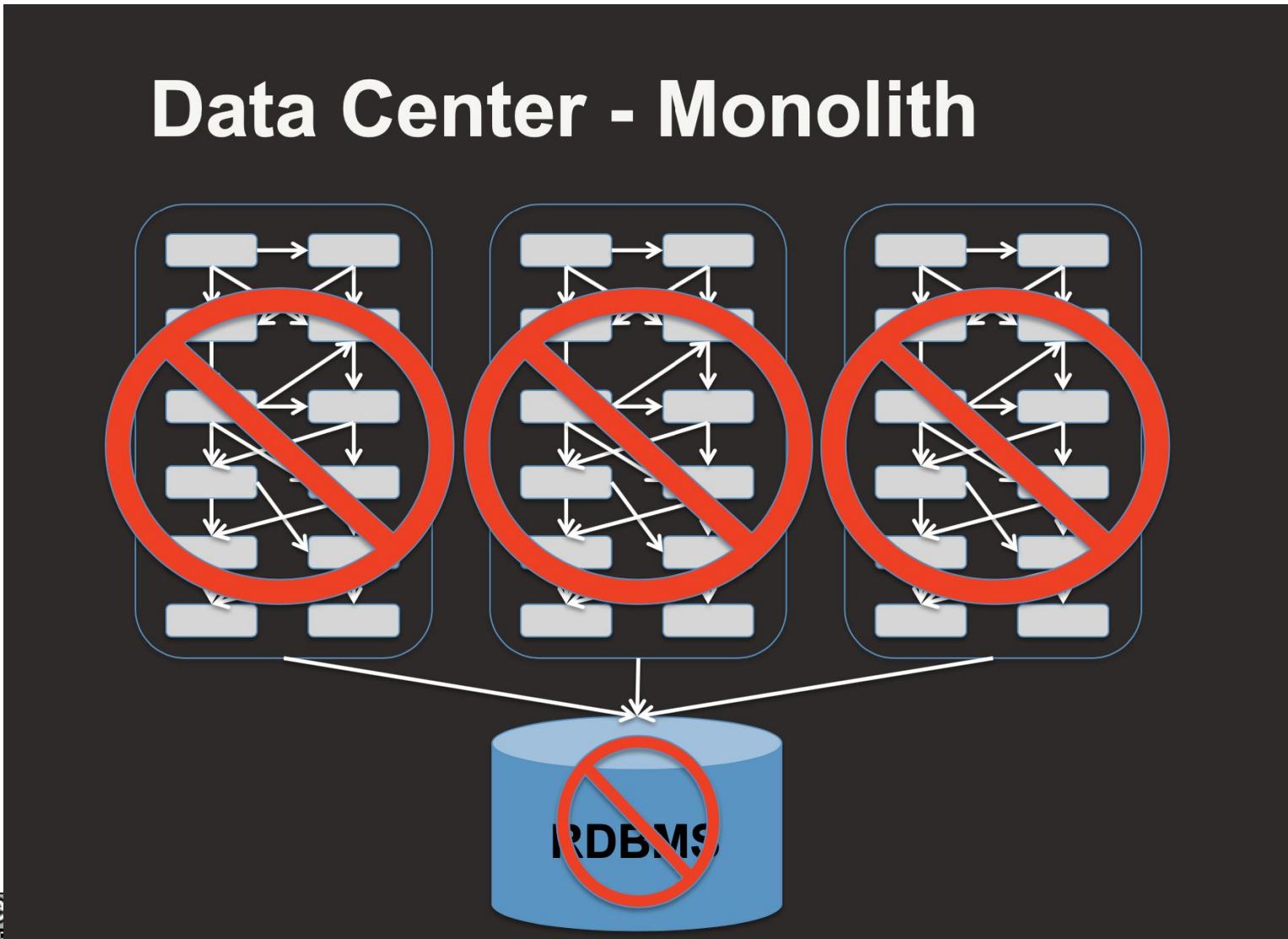
- “Each service has a team associated with it, and that team is completely responsible for the service—from scoping out the functionality, to architecting it, to building it, and operating it... **You build it, you run it**” Werner Vogels, CTO, Amazon
 - Promotes Customer Focus and Innovation
 - Gives developers direct access to customers
 - And experience of how their code performs





© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Netflix too....



August 2008



[Your Account](#) | [Queue](#) | [Help](#)

We're Sorry DVD Shipments Are Delayed

Dear Betsy,

Our shipping system is unexpectedly down. We received a DVD back from you and should have shipped you a DVD, but we likely have not. Our goal is to ship DVDs as soon as possible, and we will keep you posted on the status of your DVD shipments.

We are sorry for any inconvenience this has caused. If your DVD shipment is delayed, we will be issuing a credit to your account in the next few days. You don't need to do anything. The credit will be automatically applied to your next billing statement.

Again, we apologize for the delay and thank you for your understanding. If you need further assistance, please call us at 1-888-638-3549.

-The Netflix Team

Service



<http://www.flickr.com/photos/yjv>



© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Tightly coupled

- Tightly coupled systems have significant problems:
 - Errors, delays and downtime spread through the system
 - The resilience of the whole system is based on the weakest part
 - Cost of upgrading or migrating spreads
 - Hard to evaluate the useful parts from the dead weight

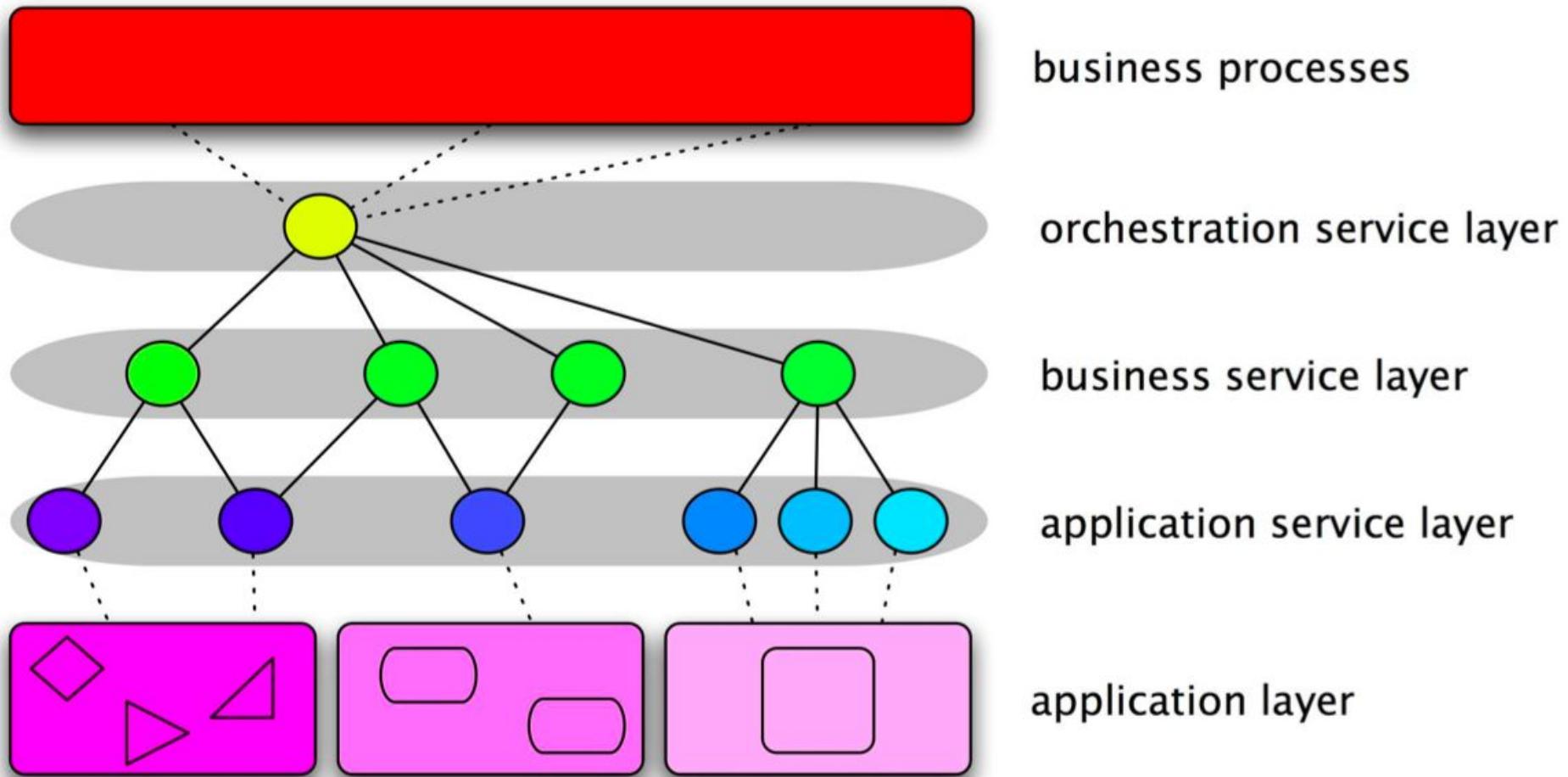


Loose coupling

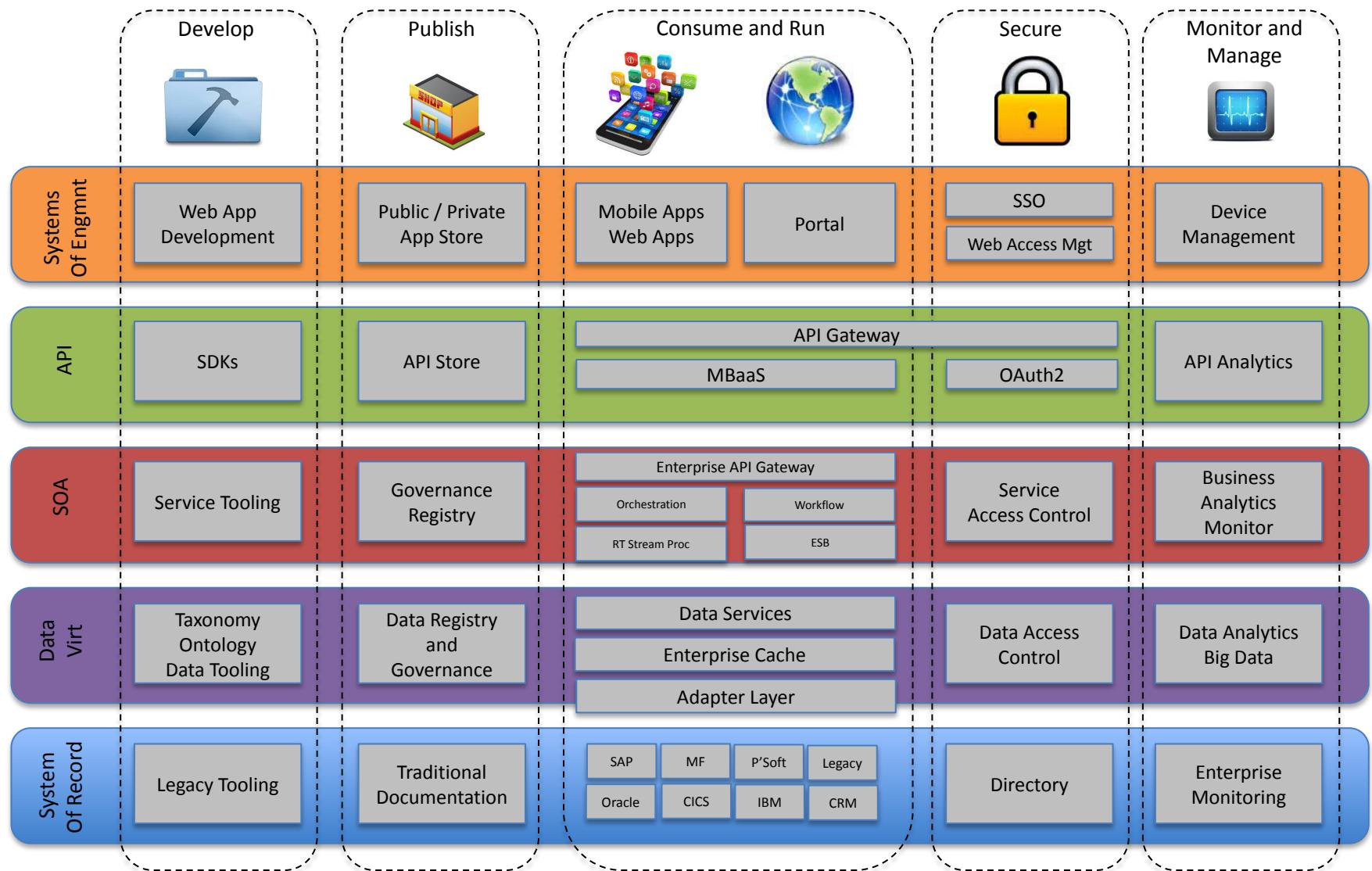


© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>

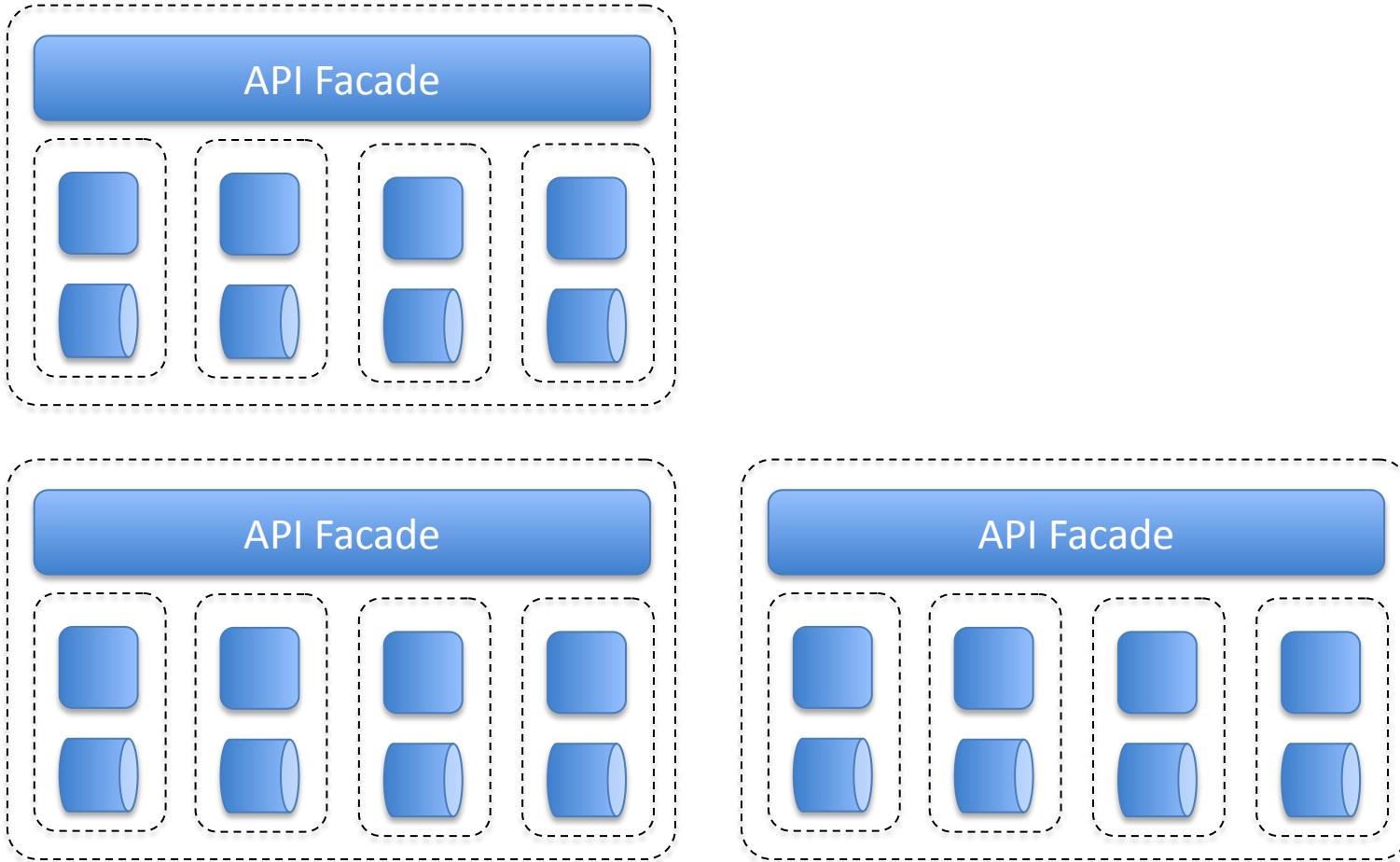
Traditional SOA = Layered Architecture



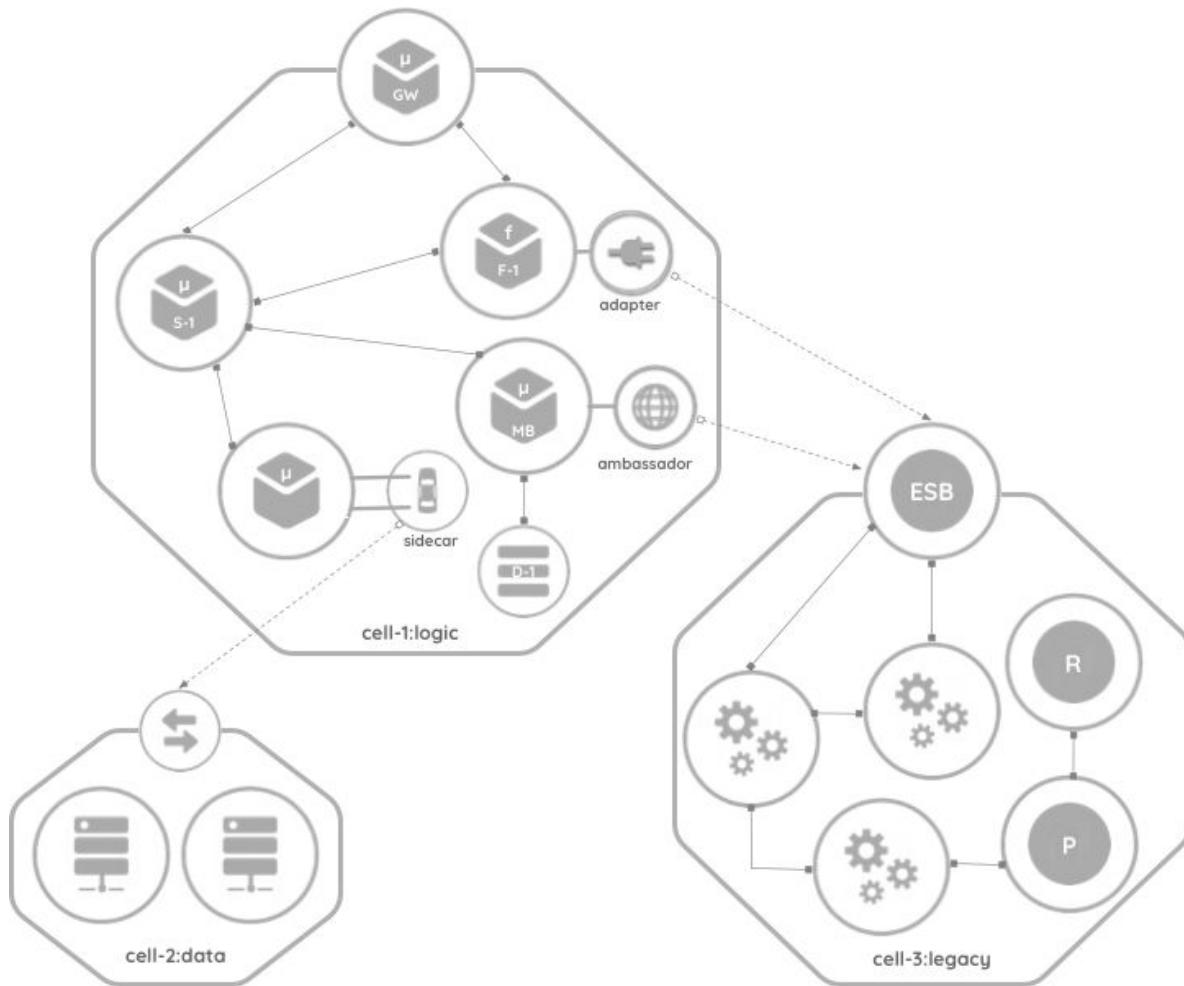
End-to-End Architecture and Functional Requirements for a Connected Business



Micro and Macro Services “Segmented Architecture”



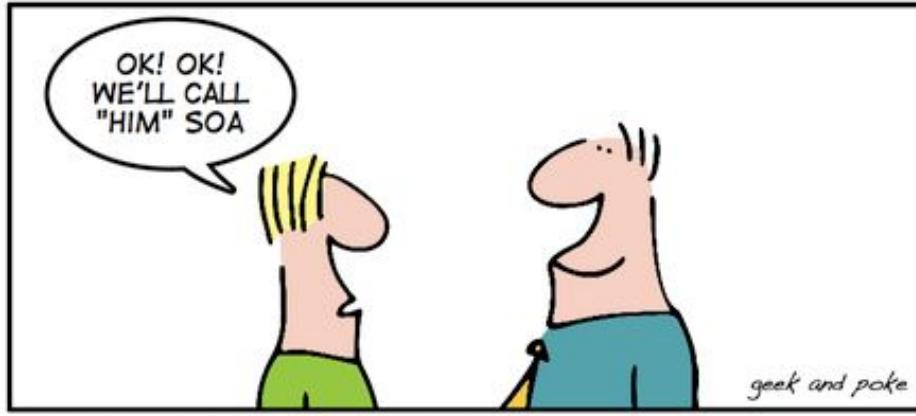
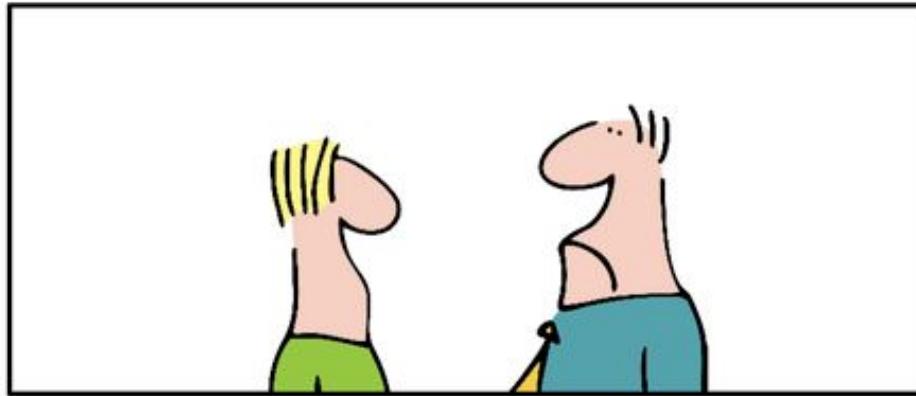
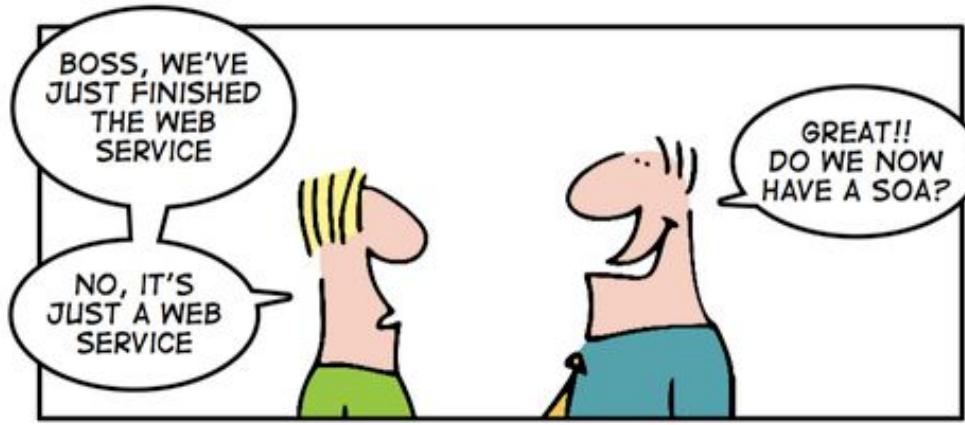
Cell Based Architecture



<https://github.com/wso2/reference-architecture/blob/master/reference-architecture-cell-based.md>



© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>

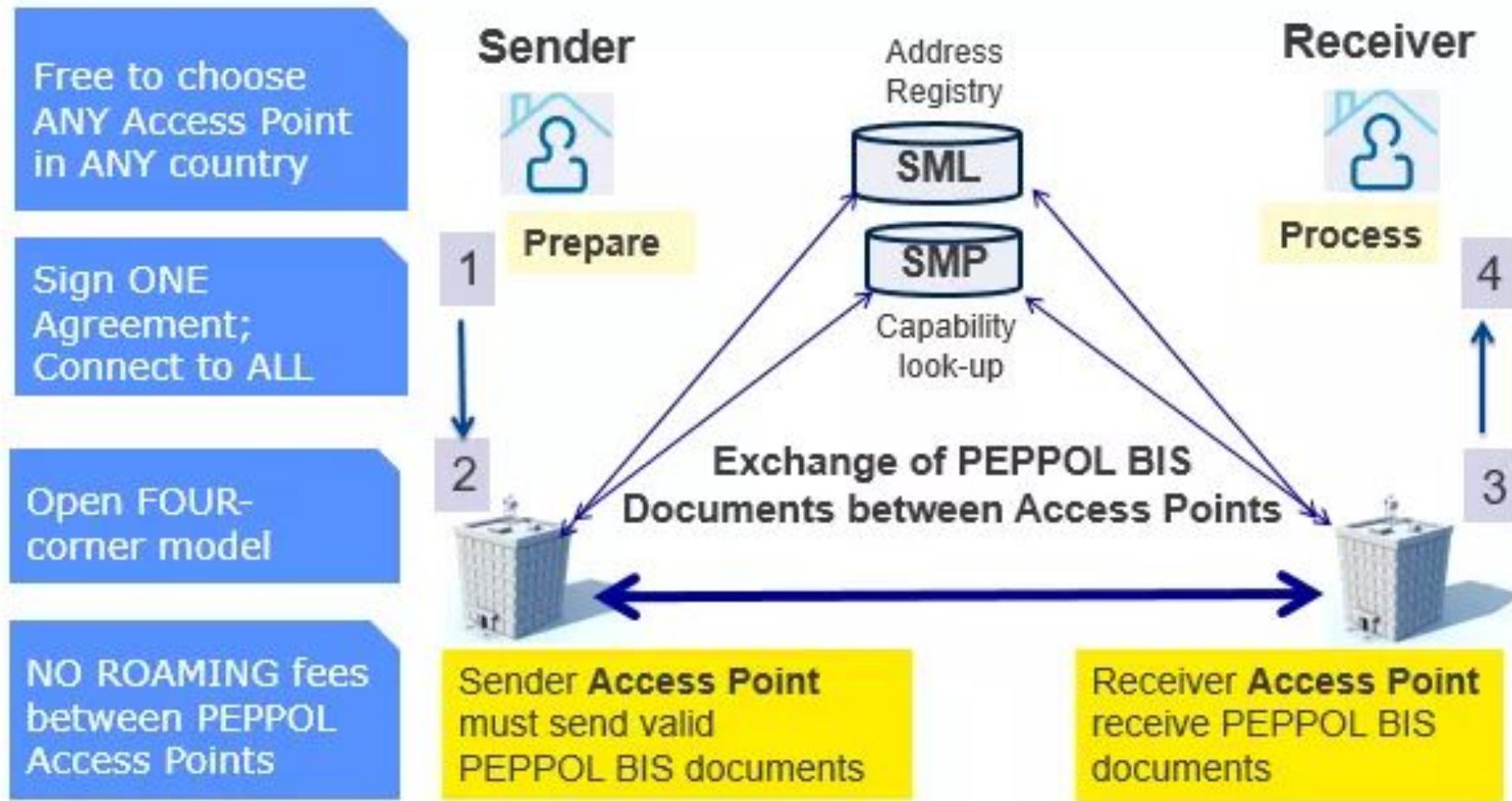


HOW TO GET A SOA



© Paul Fremantle 2C
Attribution-NonCommercial
See <http://creativecommons.org/licenses/by-nc-sa/2.0/>

PEPPOL



June 2016: Tradeshift raise \$75m
May 2018: Tradeshift raise \$250m
Jan 2020: another \$240m



WHAT IS TRADESHIFT? SOLUTIONS BY ROLES RESOURCES FREE INVOICING

ENGLISH ▾

SIGN UP LOG IN

REQUEST DEMO

Innovate the way you buy. |

Procurement, payables, supplier management—and the only open network that's simple for employees, free for suppliers, and agile enough for you.

Watch our story

Let's get started

AIRFRANCE KLM



DHL

CBRE



ZURICH

NHS
Shared Business Services



TRADESHIFT OVERVIEW

Everything you need to know about what we do. [Learn more »](#)



WHY WE'RE DIFFERENT

"You call that an open network? Seriously..." [Read me »](#)



SPEND MATTERS REVIEW

Jason Busch examines our e-procurement tool. [Download »](#)

[Home](#)[Finance & Accounting](#)[Employment Services](#)[Procurement](#)[Media & Events](#)[Contact Us](#)[About NHS SBS](#)[Our Services](#)[Working with clients](#)[Working with suppliers](#)[Supplier FAQs](#)[Postal Invoicing](#)[E-Invoicing](#)[- E-Invoicing FAQs](#)[Notices](#)[Working with GP practices](#)[Careers](#)[Feedback](#)[Business Intelligence](#)[Personal Health Budgets](#)[Home](#) ▶ [Working with suppliers](#) ▶ [E-Invoicing](#)

E-invoicing

Want to take steps to get paid quicker and submit your invoices in a way that doesn't involve paper, postage or manual handling?

e-Invoicing through Tradeshift is an exciting new business platform that is designed to make life simpler for organisations and we are working in partnership with them to deliver e-Invoicing for suppliers to our NHS clients.

Registering with Tradeshift is quick and simple and the benefits are immediate. If you are an SME or low volume supplier then the web-based portal at <http://www.tradeshift.com/supplier/nhs-sbs/> is likely to be the best solution, offering immediate access to submit and review invoices wherever you are.

If you submit a higher volume of invoices, you may wish to consider integrating your existing invoicing software with the Tradeshift system. This is a simple process, with support available, to help you see just how beneficial e-Invoicing could be for your business.

Why e-invoicing?

- **Easy setup** - it's quick and simple to get started and just as simple to use
- **Free to use** - no setup fee, transaction fees or service charges
- **Improved communication** - track the status of your invoices and run key reports
- **Instant validation** - take advantage of 15 pre-submission checks to ensure your invoice is right first time

Contact Us

T: 0303 123 1177

E: SBs-W.Payables@nhs.net

Share this

- [Delicious](#)
- [Dig It](#)
- [Facebook](#)
- [LinkedIn](#)
- [Twitter](#)

Netflix - Watch TV Programs

https://signup.netflix.com

Offline Mail Inbox (124,820) - p... WSO2 WSO2, Inc. - Calend... + bookmark Shorten with bit.ly Gmail - Inbox (720)

Questions? Call 0800 096 6379 - 24/7
Buy / Redeem Gift Member

NETFLIX

1 MONTH FREE TRIAL

Watch TV programmes & films anytime, anywhere. Only £5.99 a month.

Start Your Free Month



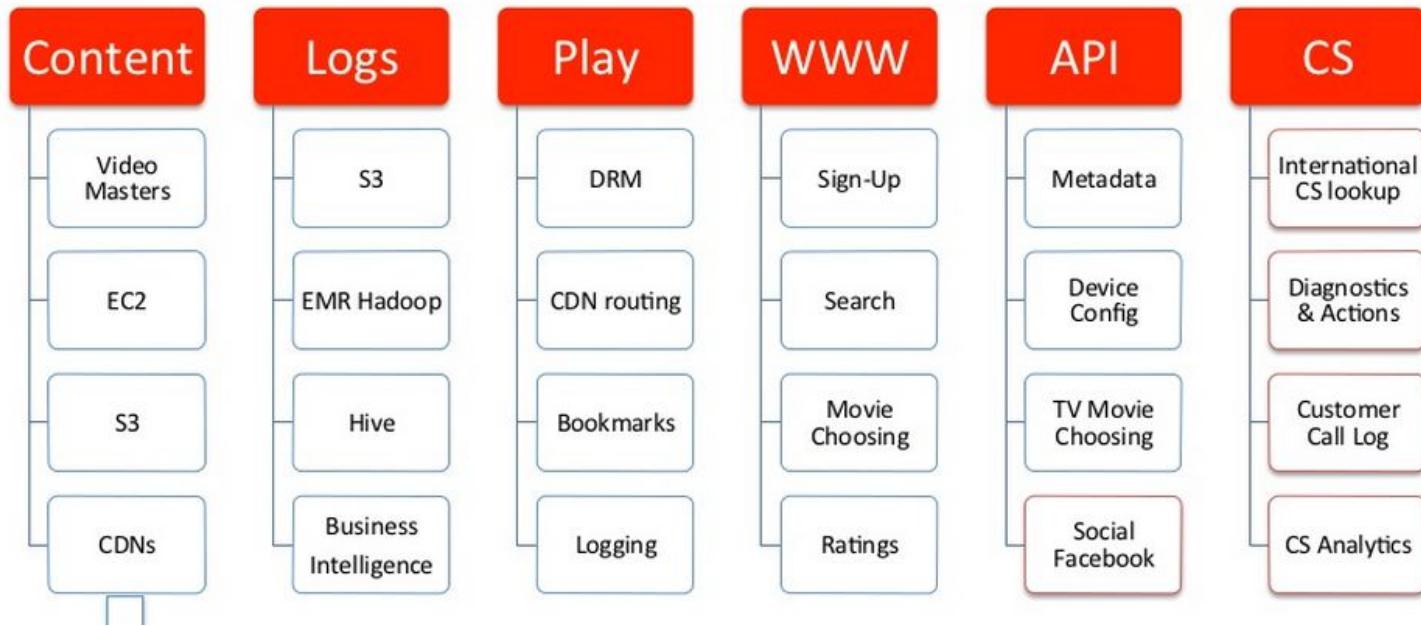
© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Netflix

- A REST and Cloud based SOA approach
- Continuous Delivery
- 100% Based in the cloud
- See excellent presentations from Adrian Cockcroft
 - e.g.
<http://www.slideshare.net/adrianco/global-netflix-platform>



Netflix Deployed on AWS



Platform Services

- Discovery – service registry for “applications”
- Introspection – Entrypoints
- Cryptex – Dynamic security key management
- Geo – Geographic IP lookup
- Platformservice – Dynamic property configuration
- Localization – manage and lookup local translations
- Evcache – eccentric volatile (mem)cached
- Cassandra – Persistence
- Zookeeper - Coordination
- Various proxies – access to old datacenter stuff



The (in)famous Chaos Monkey

- Randomly kills machines
- Yes, production systems
- Proves that the system is resilient

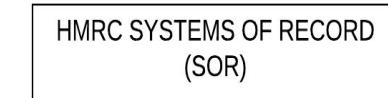
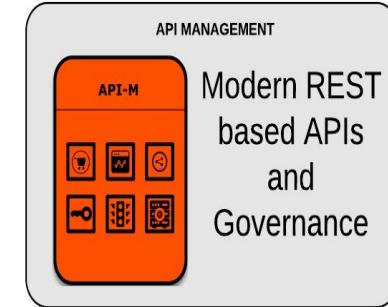


Use Case

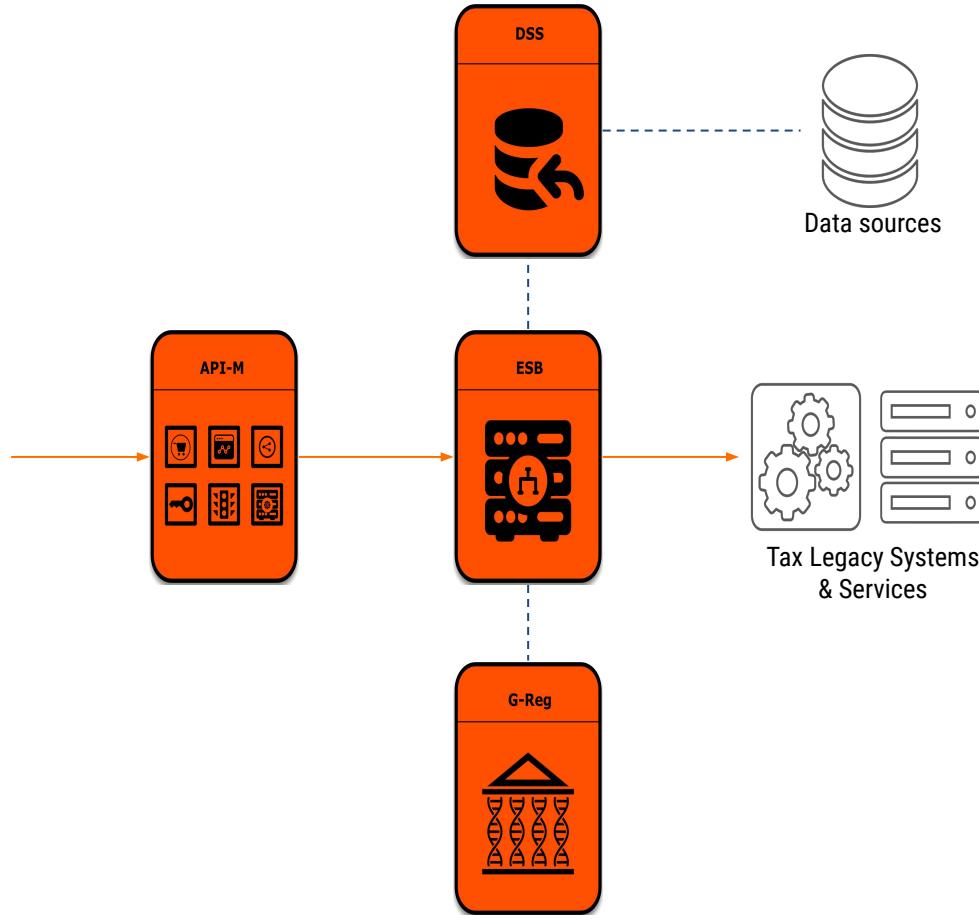
- ❖ Businesses and individuals make tax returns online without having to use older technology and paper-based alternatives.
- ❖ Via Managed APIs, online Tax Applications get access to functionality of HMRC and 3rd Party Services on the Tax Platform
- ❖ Tax Legacy Systems that maintain all tax related information of taxpayers & perform tax processing

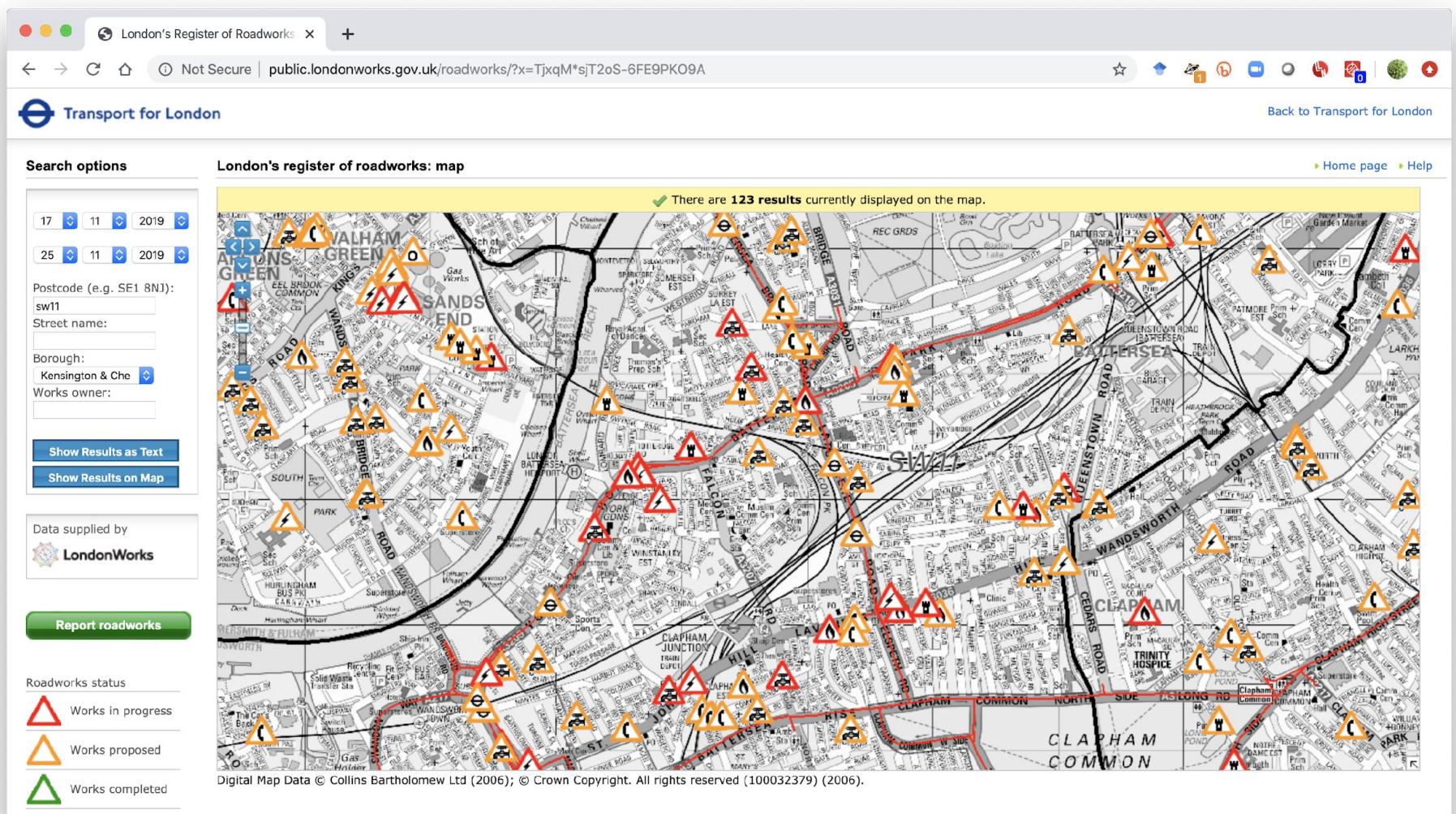


New Digital GOV.UK
Personal and Business
Tax Account management

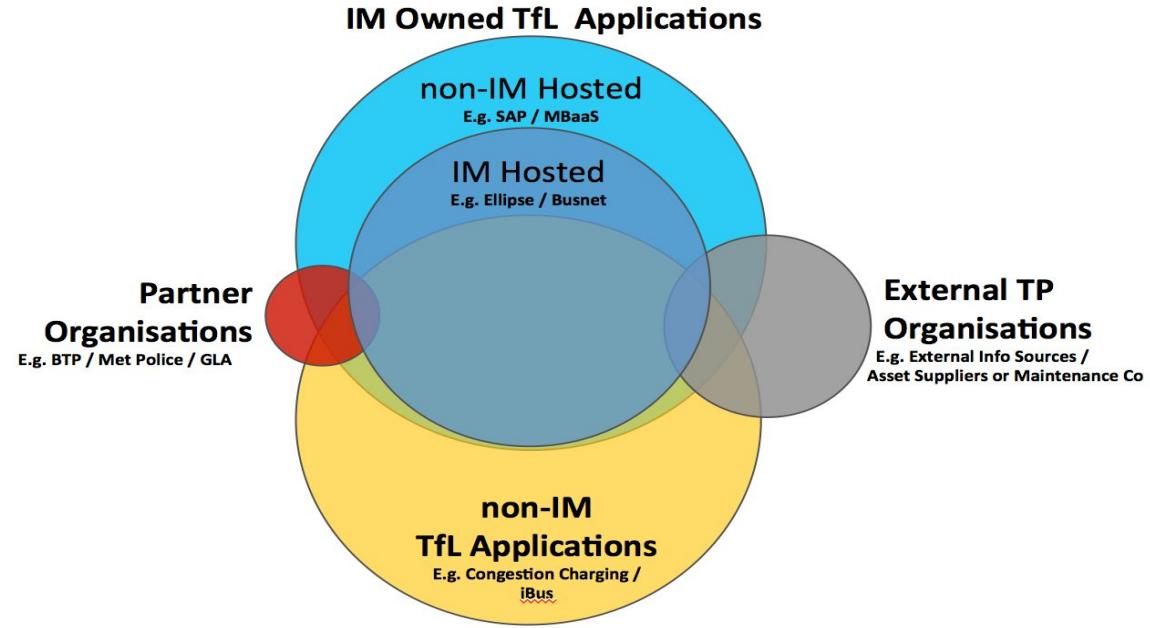


Solution Architecture





LondonWorks

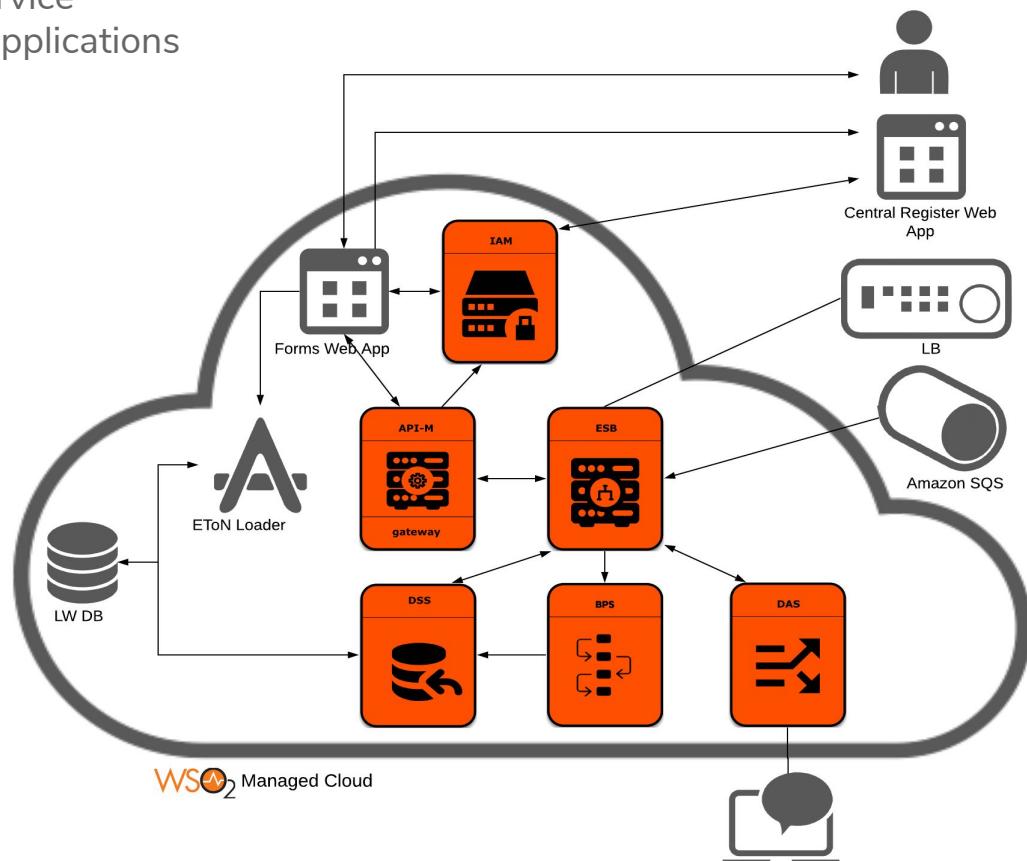


- ❖ TfL (Transport for London) is a Transportation Service Provider over 150 years.
- ❖ Around 31 million journeys are made across the TfL network everyday.
- ❖ TfL Integration Service is a service provided by IM (Information Management) department of TfL, which connects both IM and non-IM assets, and external organisations to provide/consume services to/from TfL.
- ❖ LondonWorks is a collection of applications used to manage the planning, programming, and delivery of street works on London's road network. It is the definitive source of road and street works information throughout TfL.



LondonWorks Architecture

- ❖ **Identity Service** - Unified identity across TfL network
- ❖ **Integration Service** - TfL Integration Service between databases, and various other applications
- ❖ **API Manager** - API management for externally exposed APIs
- ❖ **Analytics** - Event filters and alerts management
- ❖ **Business Process** - management on TfL Integration Service



Twitter Architecture

- Open Sourced their technology:
 - Finagle
 - <http://twitter.github.io/finagle/>
 - Called an RPC system, but completely asynchronous
 - Based on “Services”



<http://monkey.org/~marius/talks/twittersystems/#4>

Late 2012 architecture

Many **open source** components

- Memcache, redis, MySQL, etc.
- Necessarily heterogeneous

Organized around **services**

- Distinct responsibilities
- Isolated from each other
- Distributed computation and data
- RPC between systems

Multiplexing HTTP frontend

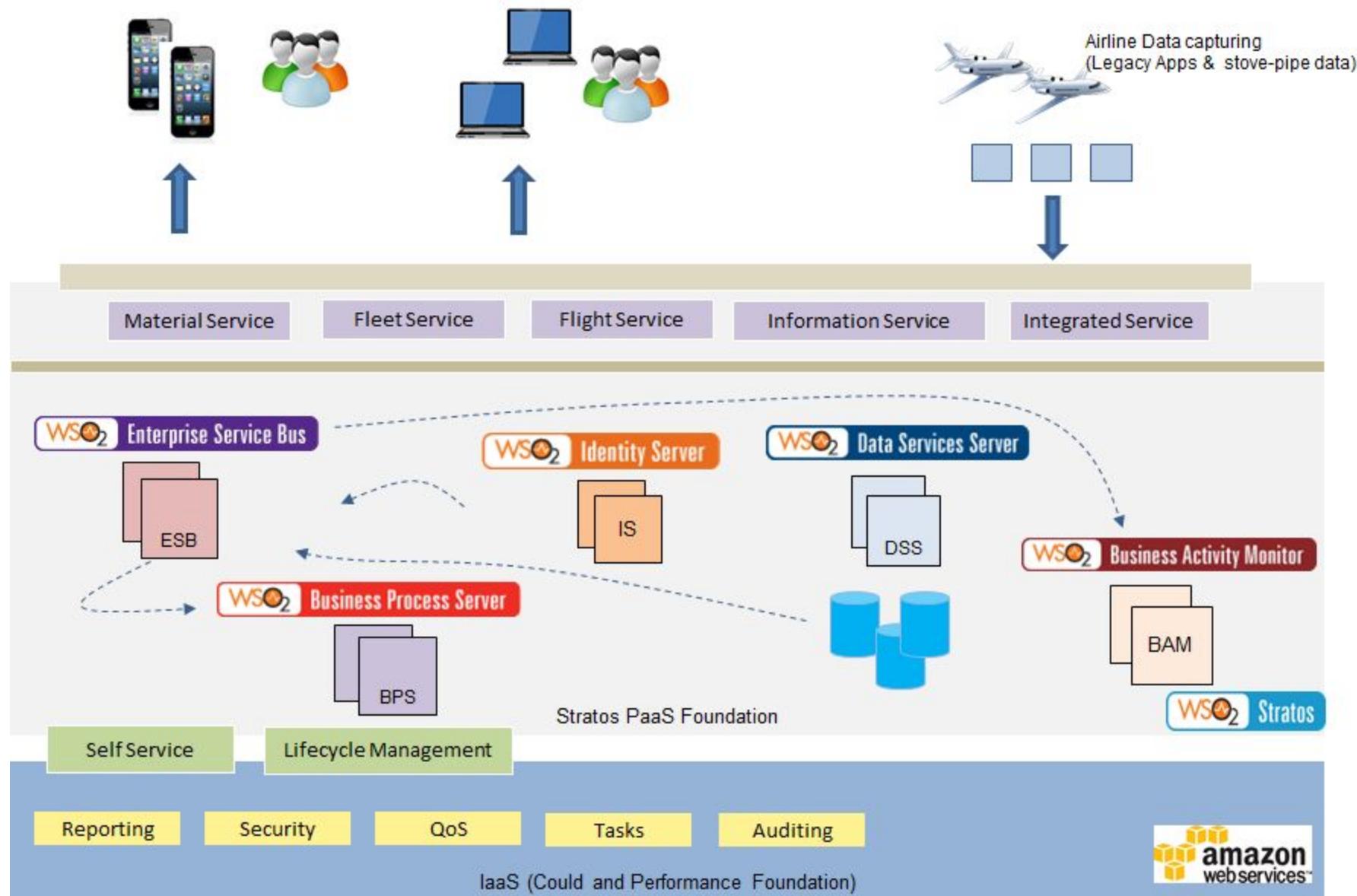
- Crucial for modularity, load balancing



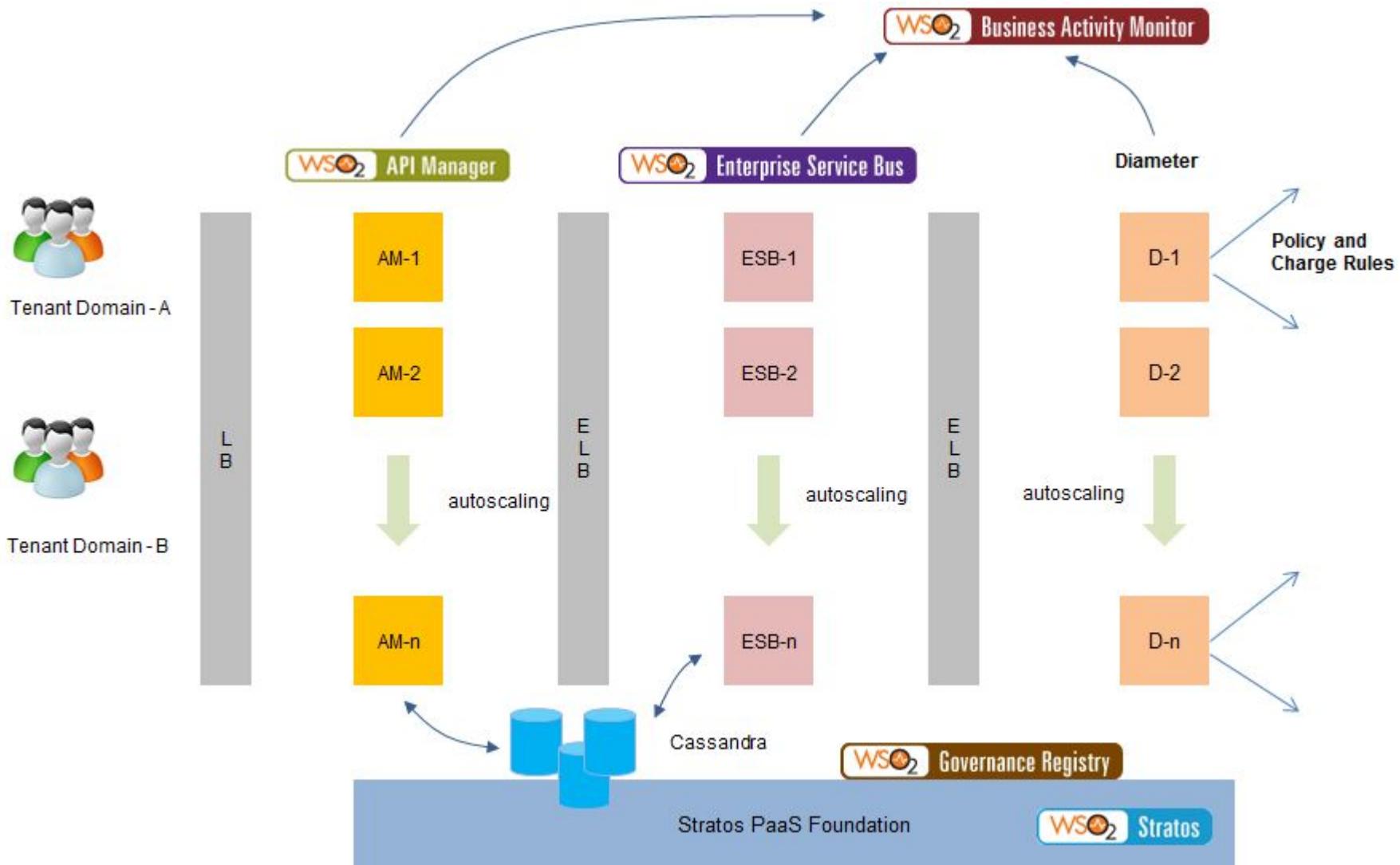
Boeing Digital Airline



Case Study : Boeing - A PaaS based Integration and API ecosystem



Case Study : Multi-tenanted Mobile Orchestration Gateway Platform

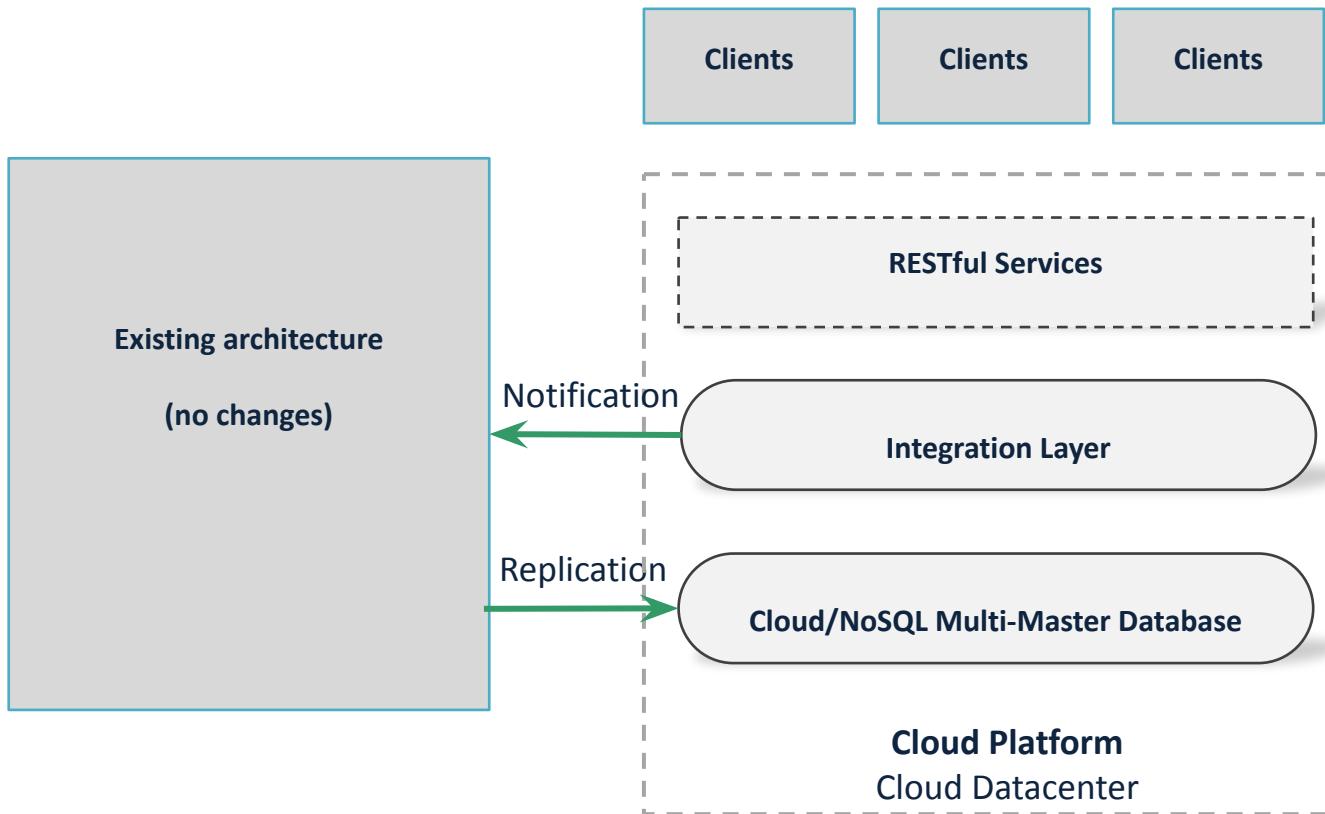


Pay TV company

- Needed to scale up to provide instant pay-as-you-go on mobile devices
- Support Disaster Recovery (DR)
- Elastic Scale e.g. during an important football match



Architecture



Anti-patterns

- Use a full waterfall model
- Don't budget time for integration test
 - Assume that standard coding unit test->integration test will work
- Build unit tests that don't test interoperability
 - E.g. Simulate XML request/response inside the calling system rather than calling a remote system
- Wait until all the systems are ready before starting any integration test
 - A delay to one system will hold up testing all the others
- Don't bother with continuous build and test
 - Even better build by hand
 - **Even better** test by hand too
- Have a nice complex process to hand over from development to test
 - That way each defect will take a long time
- Wait until the project is failing to find out your team doesn't have the skills



Questions?



© Paul Fremantle 2016 except where credited elsewhere. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License
See <http://creativecommons.org/licenses/by-nc-sa/4.0/>