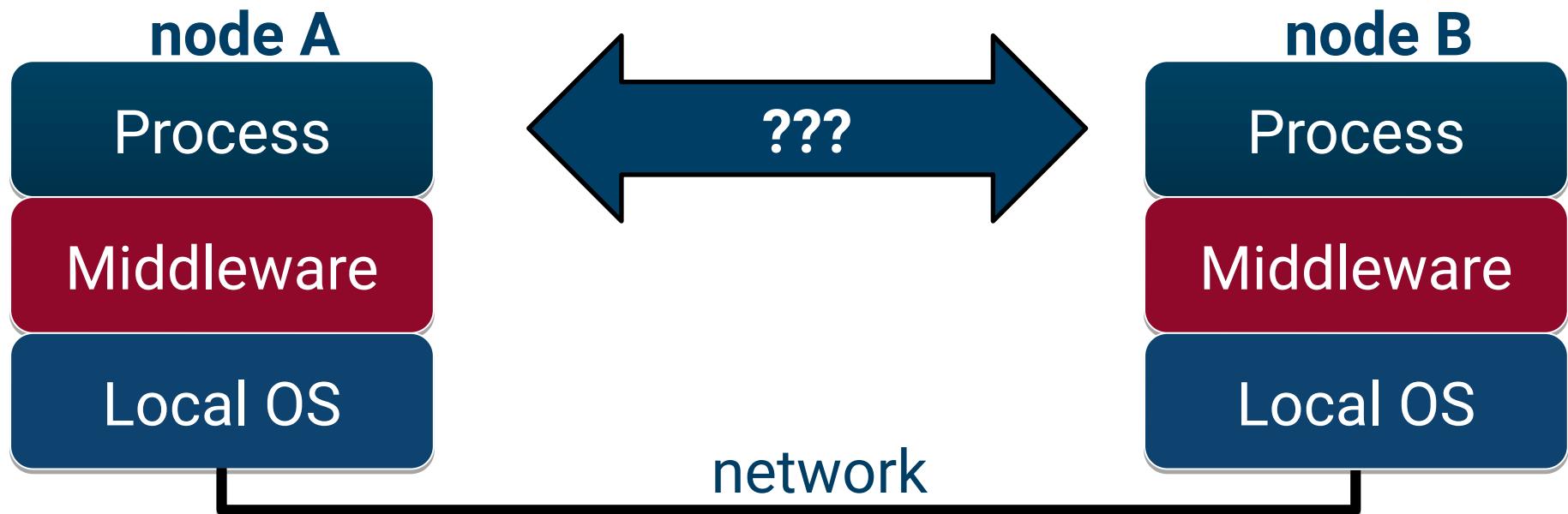


Service Oriented Architectures (SOAs)

Recap from distributed systems

A distributed system is a system where

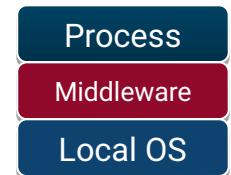
- hardware and software components are located at **networked** computers
- components communicate and coordinate their actions **ONLY by passing messages**



What we saw up till now:
Basic process to process communication
✉ How does this scale?

Let's zoom out...

distributed application



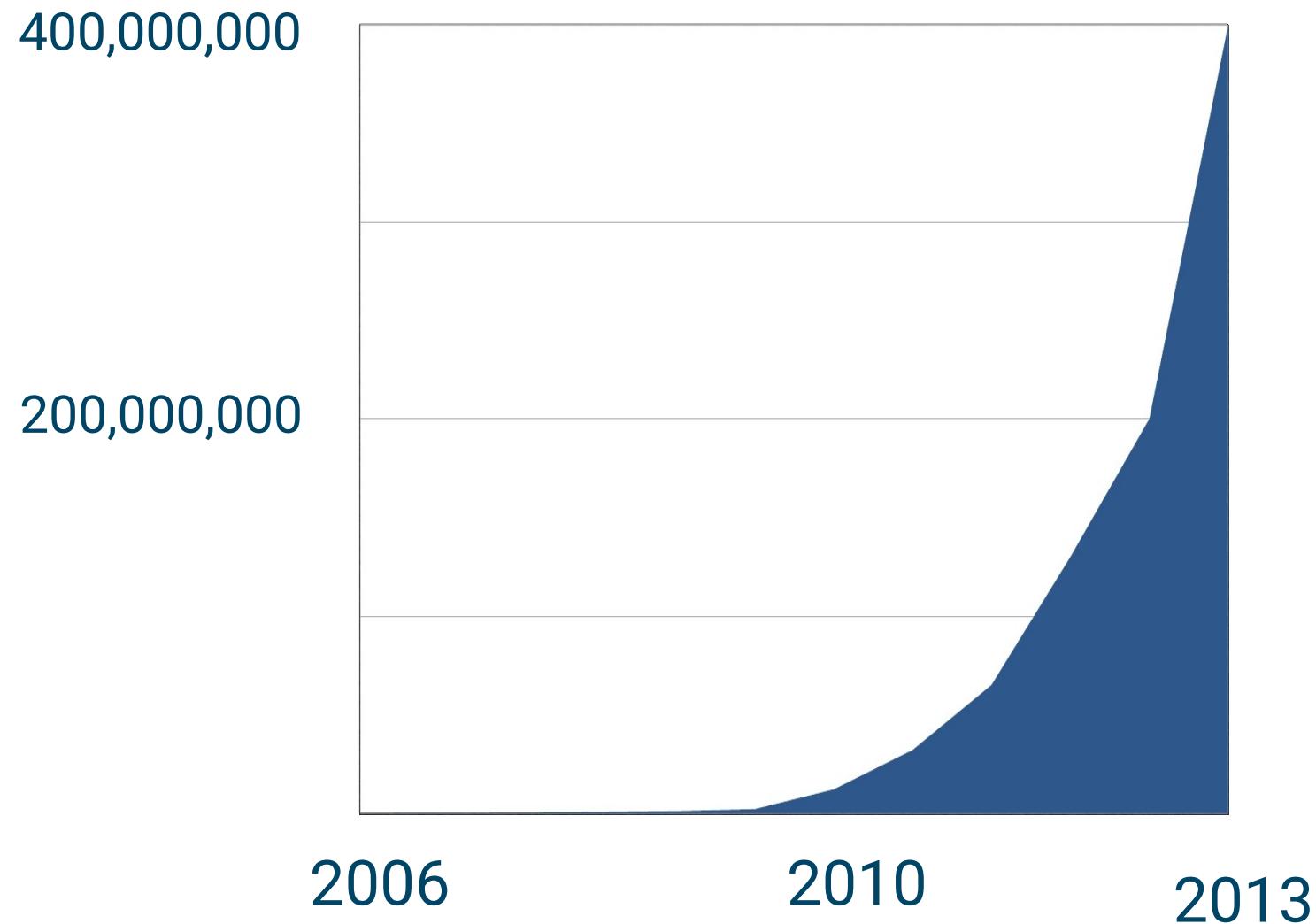
Today's distributed applications can have up to hundreds of thousands of servers

How do you organize the processes?



How Twitter actually did it...

Tweets per day



Routing

Presentation

Logic

Storage

Monorail



MySQL

Routing

Presentation

Logic

Storage

Monorail



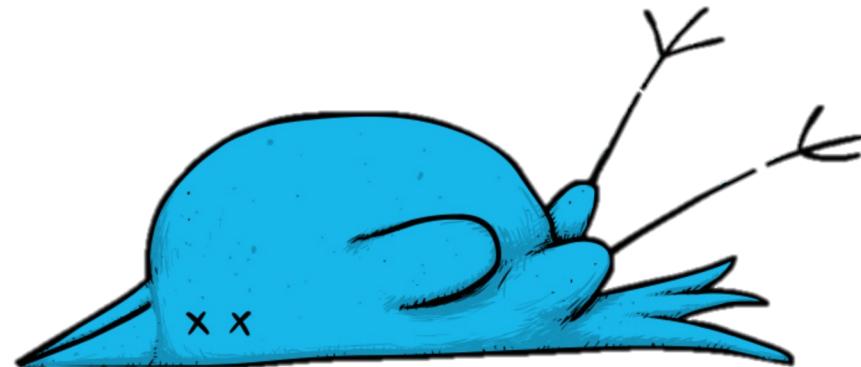
MySQL

Q: any potential weakness?

Challenges

- Storage I/O Bottlenecks
- Poor concurrency, runtime performance
- Too many cooks in the same kitchen
- Lack of clear ownership
- Too tight coupling

#FAIL



Routing

TFE

Presentation

HTTP

Monorail

API

Web

Search

Feature-X

Feature-Y

Logic

THRIFT

Tweet Service

User Service

Timeline

Direct Message Service

SocialGraph Service

THRIFT*

Storage

MySQL

Tweet Store

Flock

User Store

Cache

Memcached

Redis

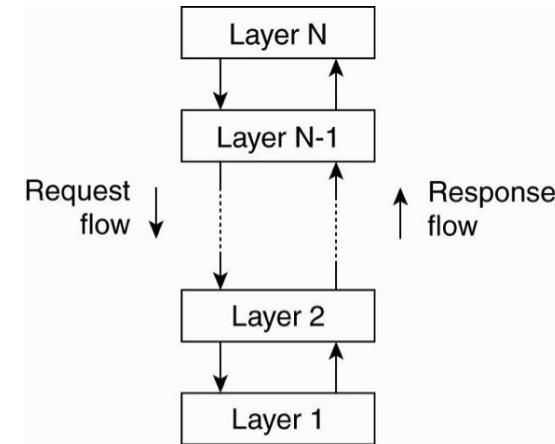
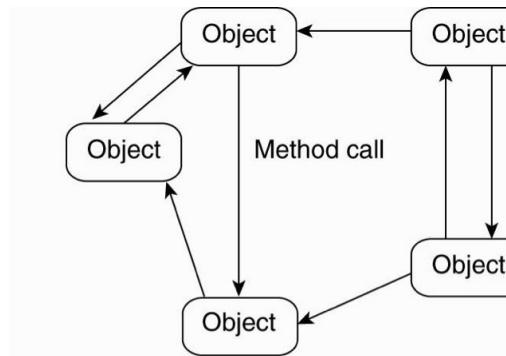
Why Twitter did what it did...

Followed the paradigm of SOAs

Tightly vs loosely coupled

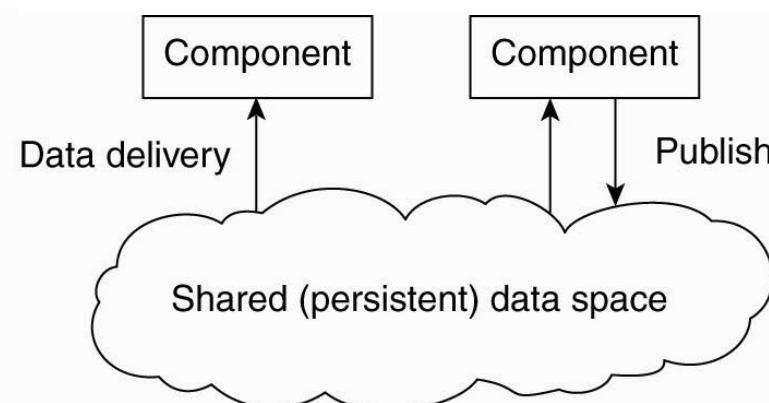
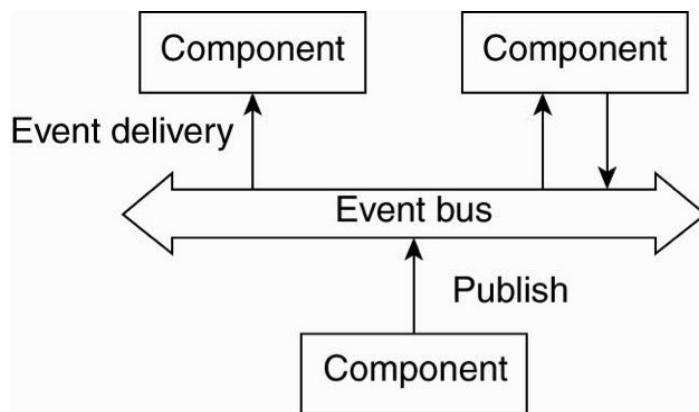
Coupled architectures:

- Components are **tightly** linked with each other
- Removing/adding a component is **non-trivial**



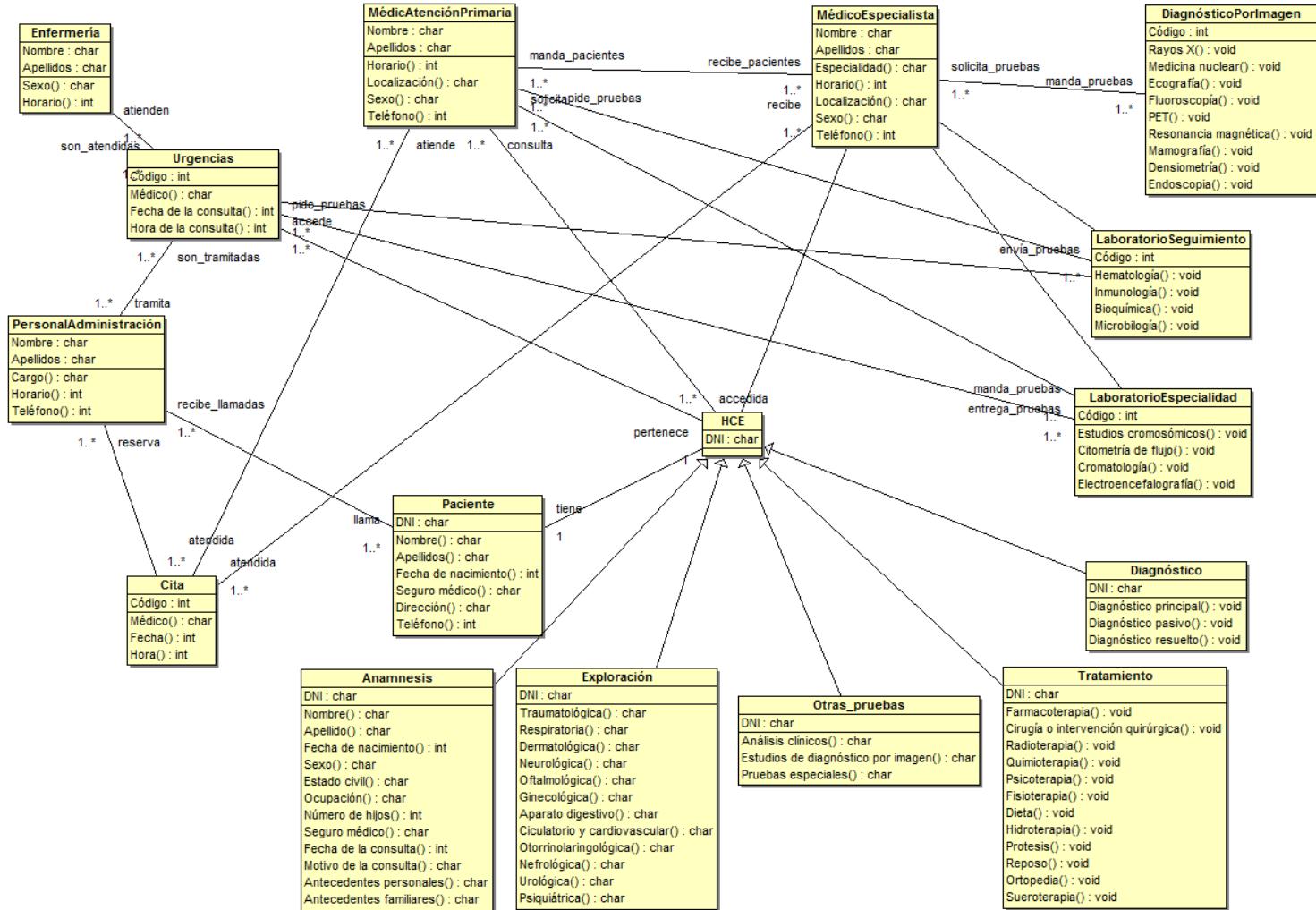
De-coupled architectures:

- Components are **loosely** linked with each other
- Removing/adding a component is **easier** and can happen **frequently**



Similarities with object-orientation

OO-design: class level
 Distributed design: process level



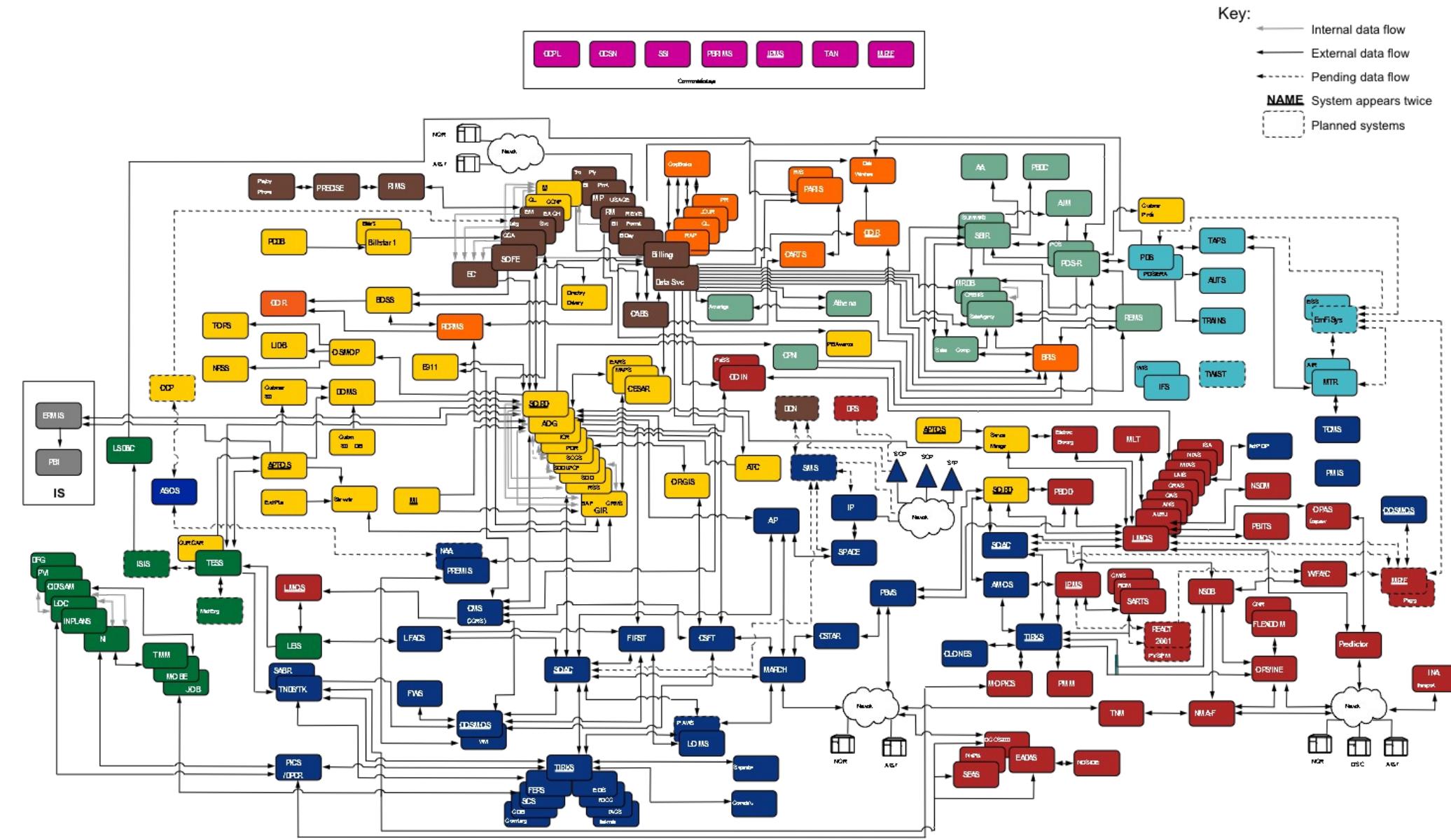
How they did it in the old days...

Distributed computing 1.0



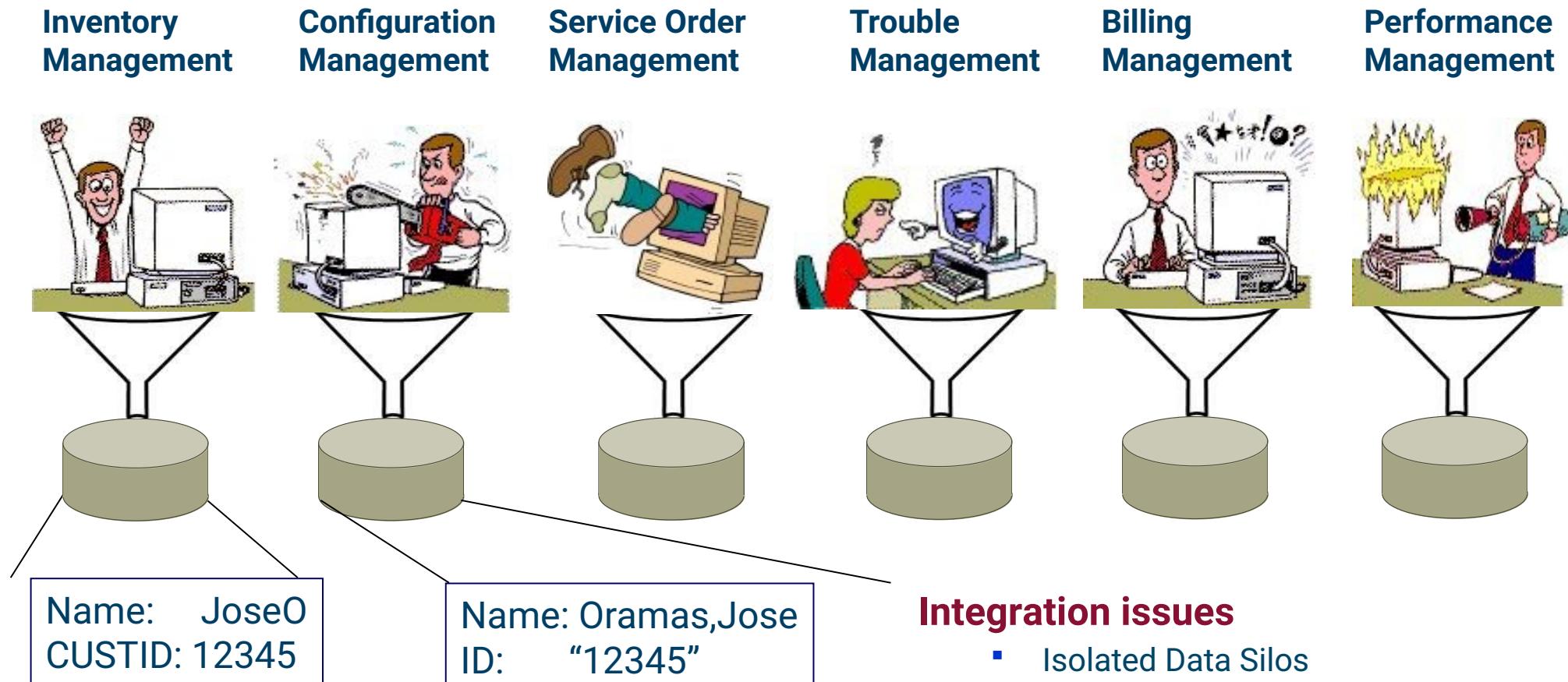
A HISTORICAL distributed system in practice

A traditional Operating Support System
= System for operating a communications network



Implications on flexibility

The result is a stovepipe system



Architectural issues

- Data redundancy
- Synchronization problems
- Vendor and Application “lock in”

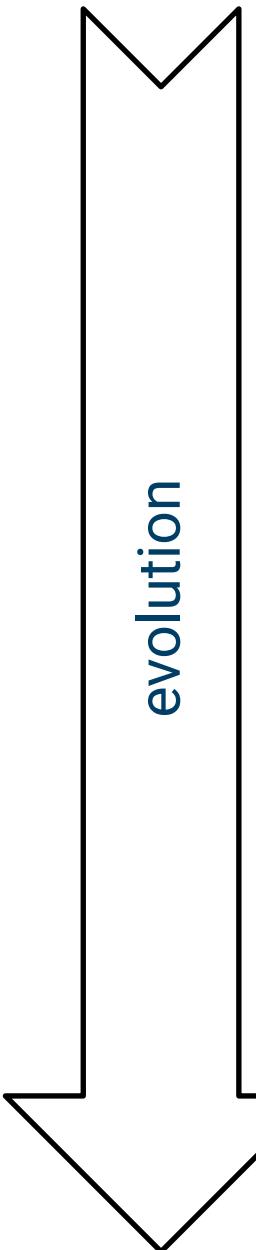
Integration issues

- Isolated Data Silos
- Administrative nightmare
- Integration/customization nightmare

We need a new approach



Link with software engineering approaches



Object-oriented design

- Based on mental model of actual or imagined objects
- Modelling real-world environments

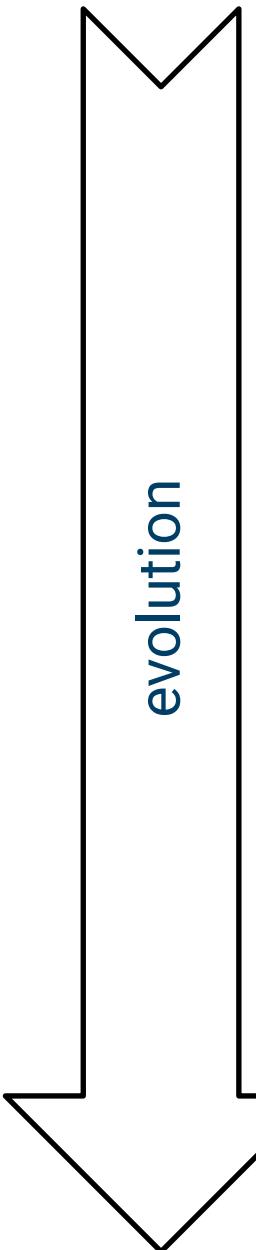
Component-oriented design

- Focus on separation of concerns
- Split into loosely coupled reusable components

Service-orientation

- Foundations in component-oriented design
- Component = service
- Service is often interchangeable & stand-alone

Link with software engineering approaches



Object-oriented design

- Based on mental model of actual or imagined objects
- Modelling real-world environments

Component-oriented design

- Focus on separation of concerns
- Split into loosely coupled reusable components

Service-orientation

- Foundations in component-oriented design
- Component = service
- Service is often interchangeable & stand-alone

How modern distributed systems are build nowadays

Distributed computing 2.0



2 Key questions....

1) What is service orientation?

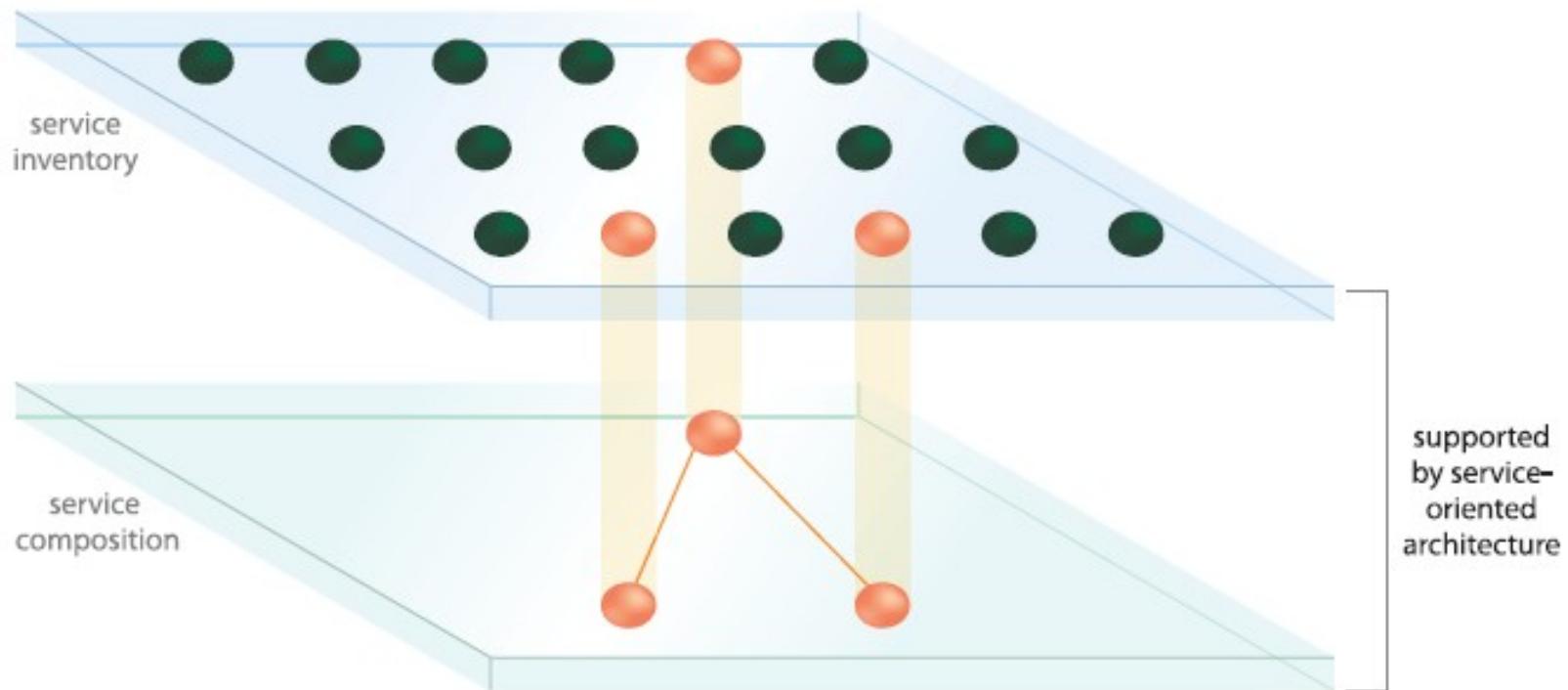
service oriented
architectures

**2) What is an architecture?
What is it not?**

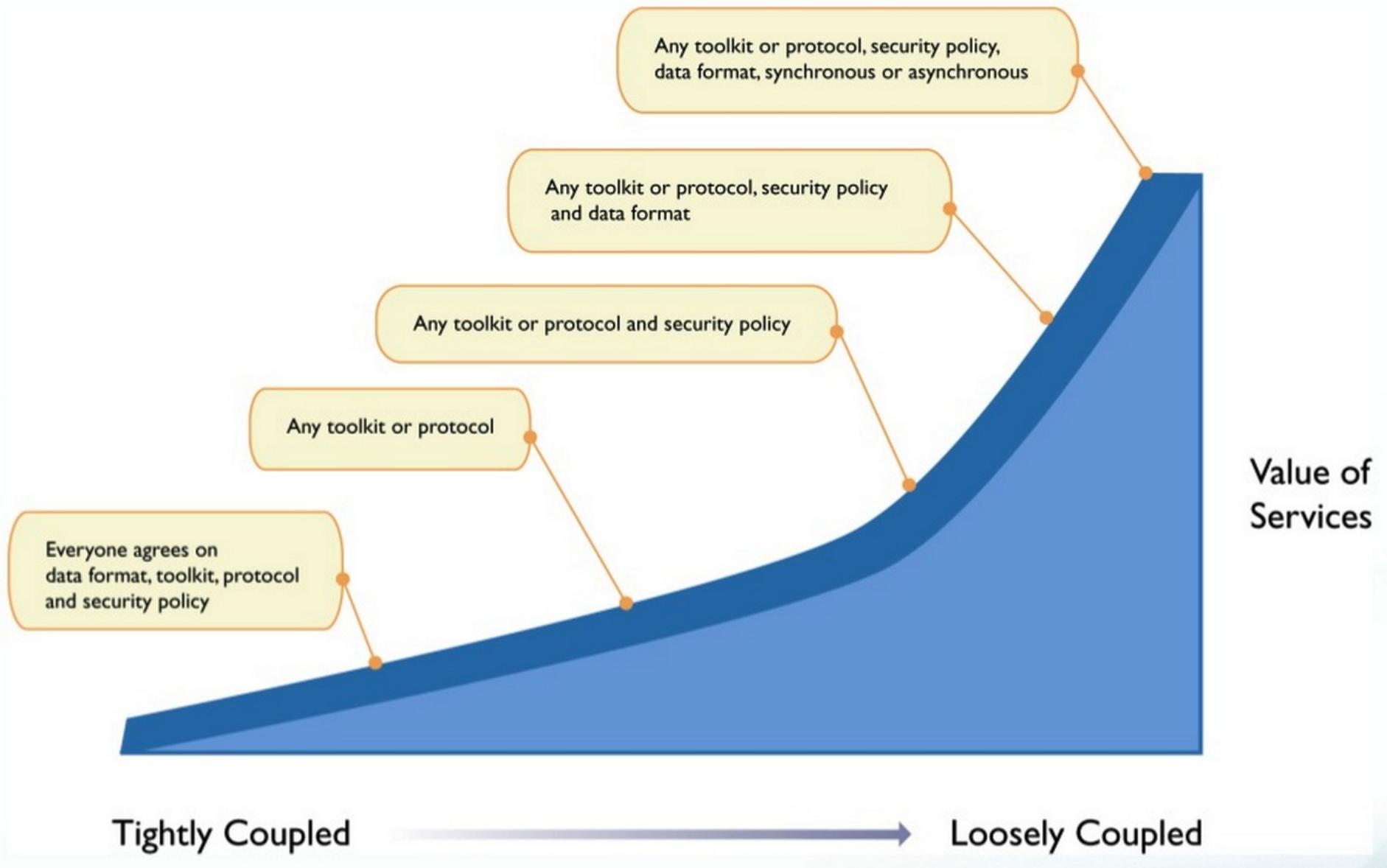
What is a service?

- Physical independent software program
- Loosely coupled
- Distinct design characteristics
- Capabilities are often defined in terms of a contract

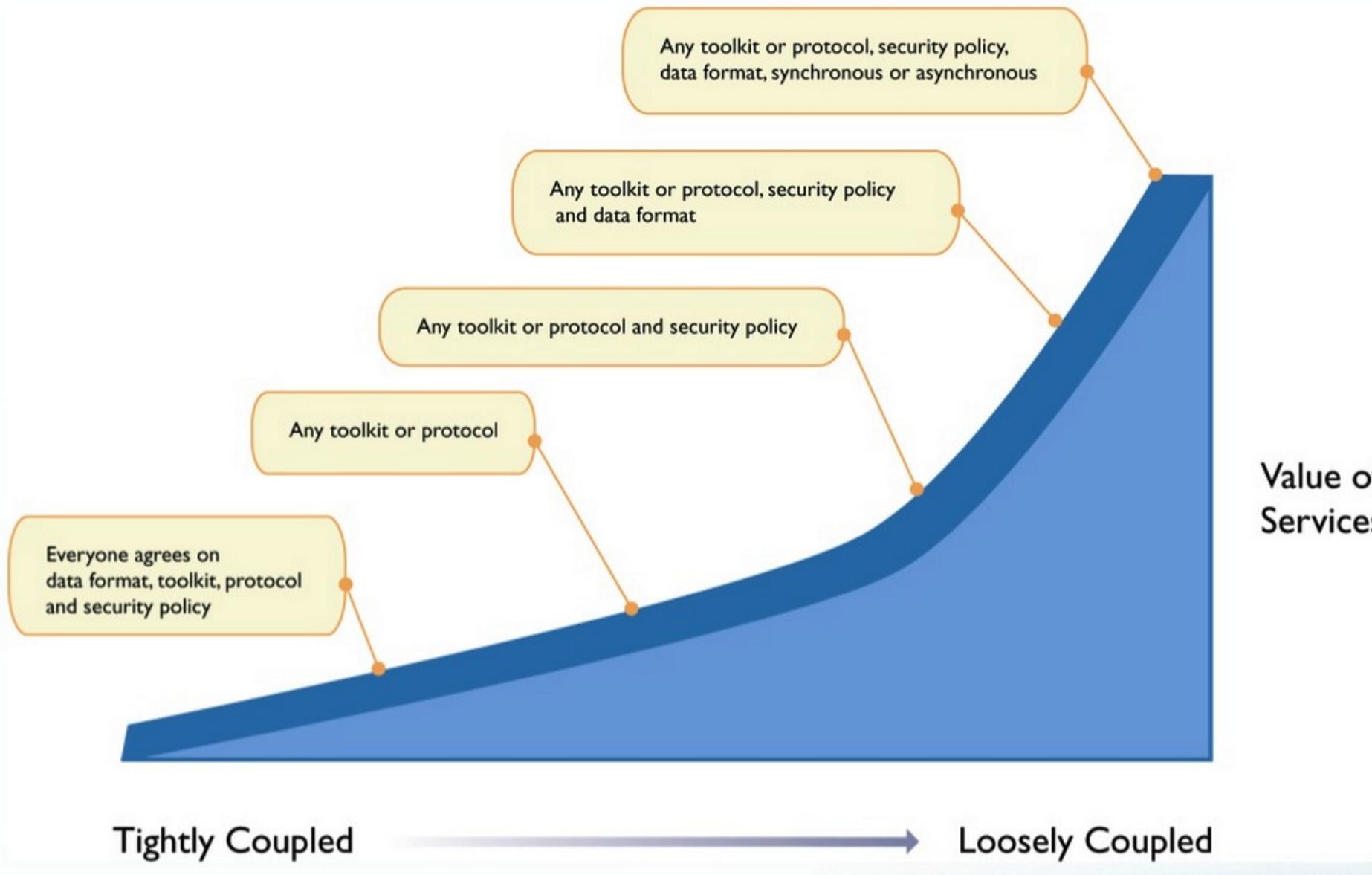
General idea: Service composition and inventory



Coupling of services & value



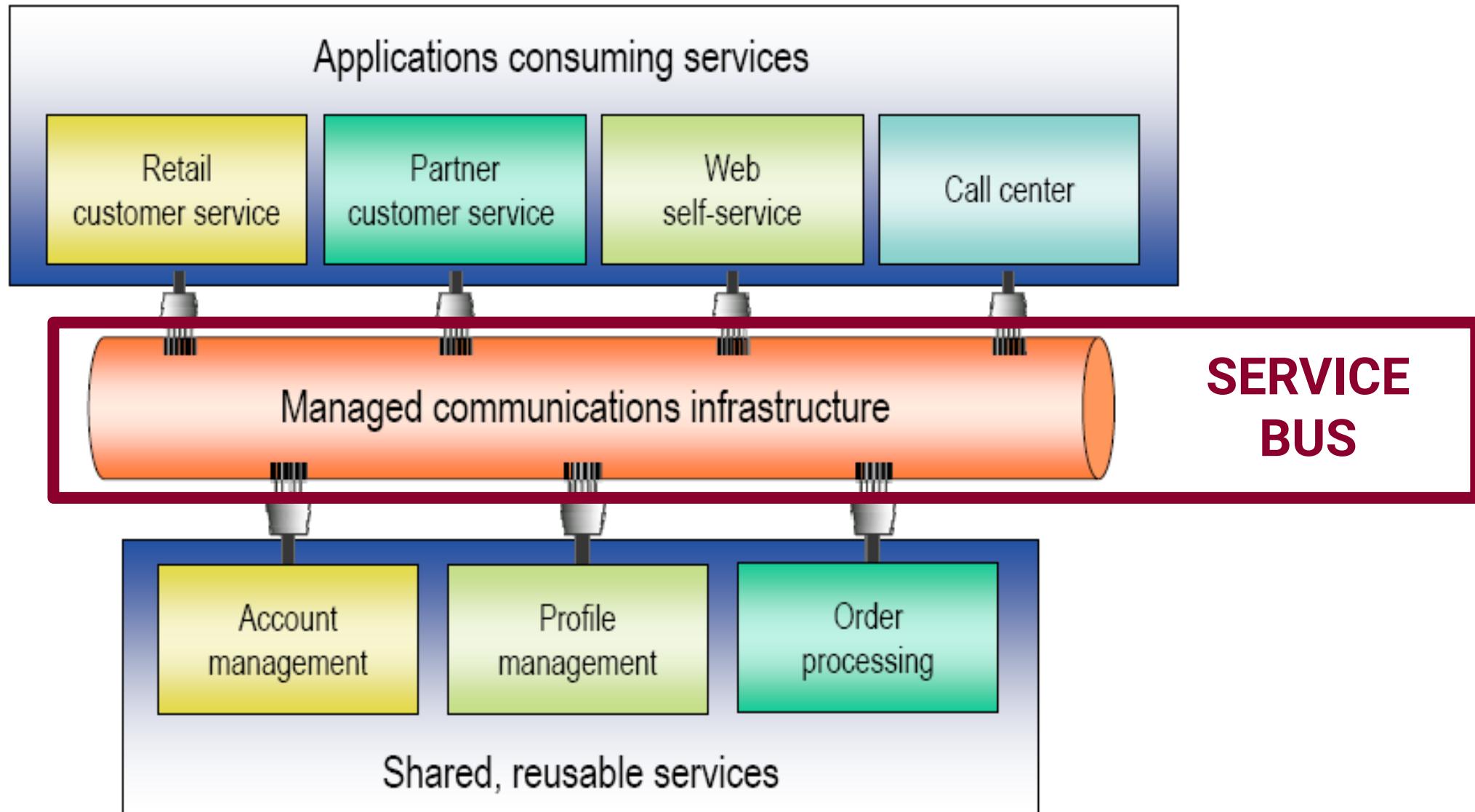
Coupling of services & value



But the glue is important...

Example of a SOA application

Focus on shared, reusable, loosely coupled services



2 Key questions....

1) What is service orientation?

service oriented
architectures

2) What is an architecture?
What is it not?

What is an SOA?

Service oriented architecture

- Architecture = **style of design**
 - Enterprise architectural style  **Technology neutral!**
- Service-oriented: the core unit of design is a service
 - Implementing a capability so that it can be easily consumed
- An approach to designing systems, a set of design principles
- A way of thinking about a problem, a mindset

From a technical perspective

- An architecture for designing systems
 - A “service” **exposes a discrete capability**
 - Any application that needs the capability uses the service
- A service exposes its capability **via an interface**
- Compared against an application-centric design perspective
 - Monolithic application silos
 - Duplication of functionality

Service Oriented Architecture (SOA)

*“A **paradigm** for **organizing** and **utilizing distributed capabilities** that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.”*



OASIS Reference Model for Service Oriented Architecture 1.0,
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm

Technical realization of a SOA...

Service Architecture != Product

Enterprise Java Beans: Powerful, but Java only

Technical realization of a SOA...

Service Architecture != Product

Enterprise Java Beans: Powerful, but Java only

CORBA

- Common Object Request Broker Architecture
- Independent of OS and language, but very complex
- Limited amount of vendors
- Historical

Technical realization of a SOA...

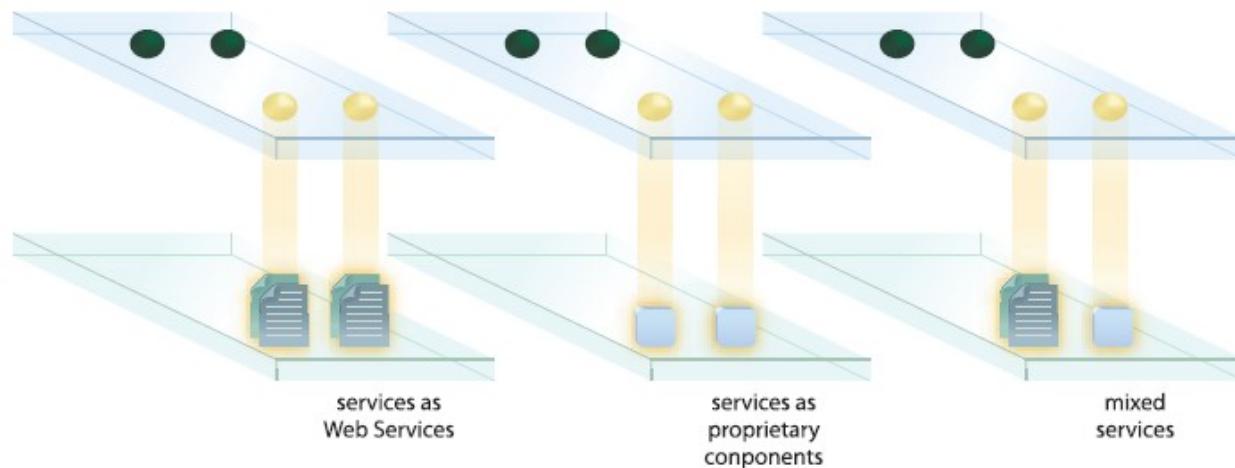
Service Architecture != Product

Enterprise Java Beans: Powerful, but Java only

CORBA

- Common Object Request Broker Architecture
- Independent of OS and language, but very complex
- Limited amount of vendors
- Historical

Web Services



Why do we need SOA?

SOA

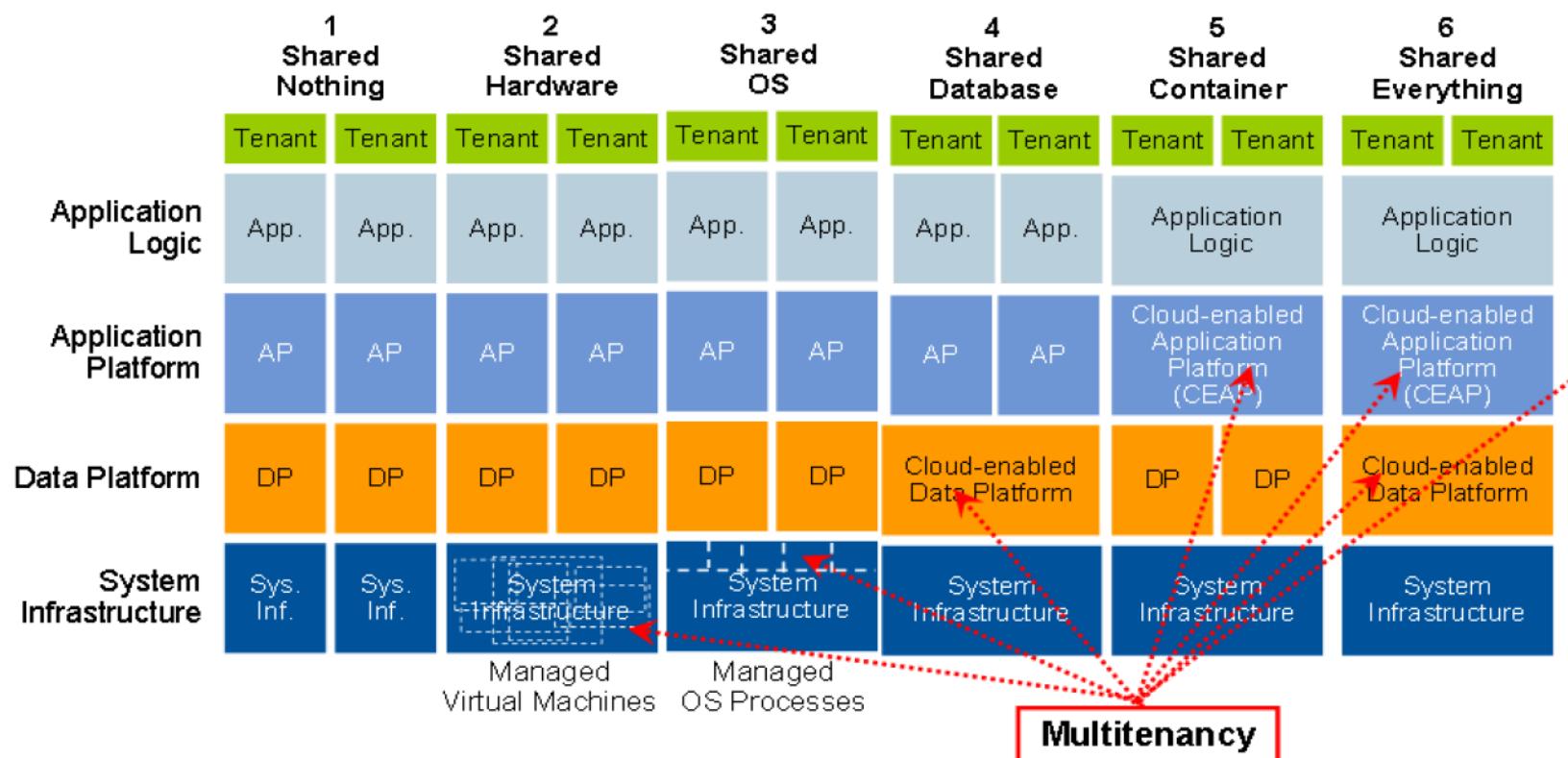
- ...promotes reusability
 - ...promotes exchangeability
- ✉ Allows multi-tenancy: a single instance of the software runs on a server, serving multiple client-organizations (tenants)

Why do we need SOA?

SOA

- ...promotes reusability
- ...promotes exchangeability

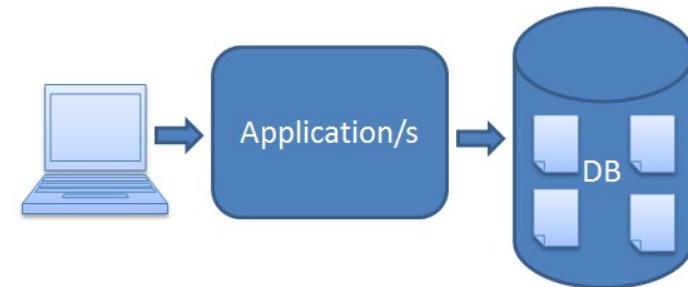
✉ Allows multi-tenancy: a single instance of the software runs on a server, serving multiple client-organizations (tenants)



Source: Gartner (August 2010)

Types of multi-tenancy

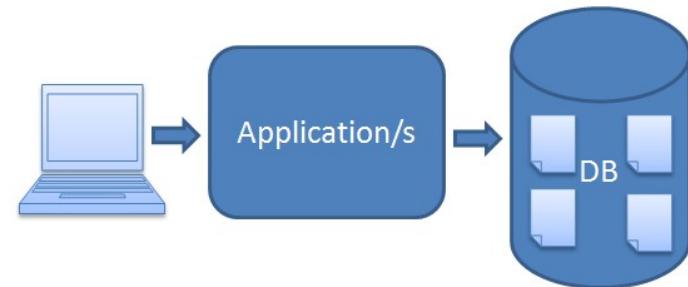
Data multi-tenancy



→ Same application + different conditions on the data

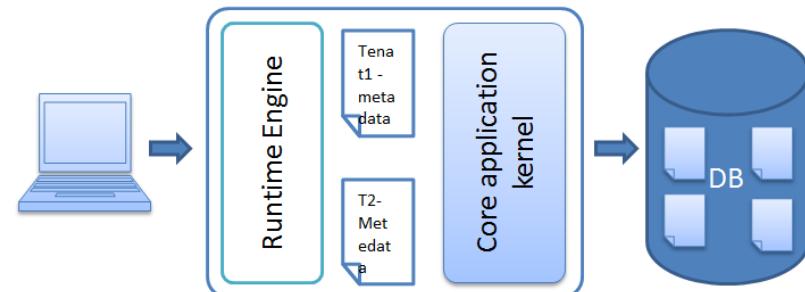
Types of multi-tenancy

Data multi-tenancy



→ Same application + different conditions on the data

Application multi-tenancy

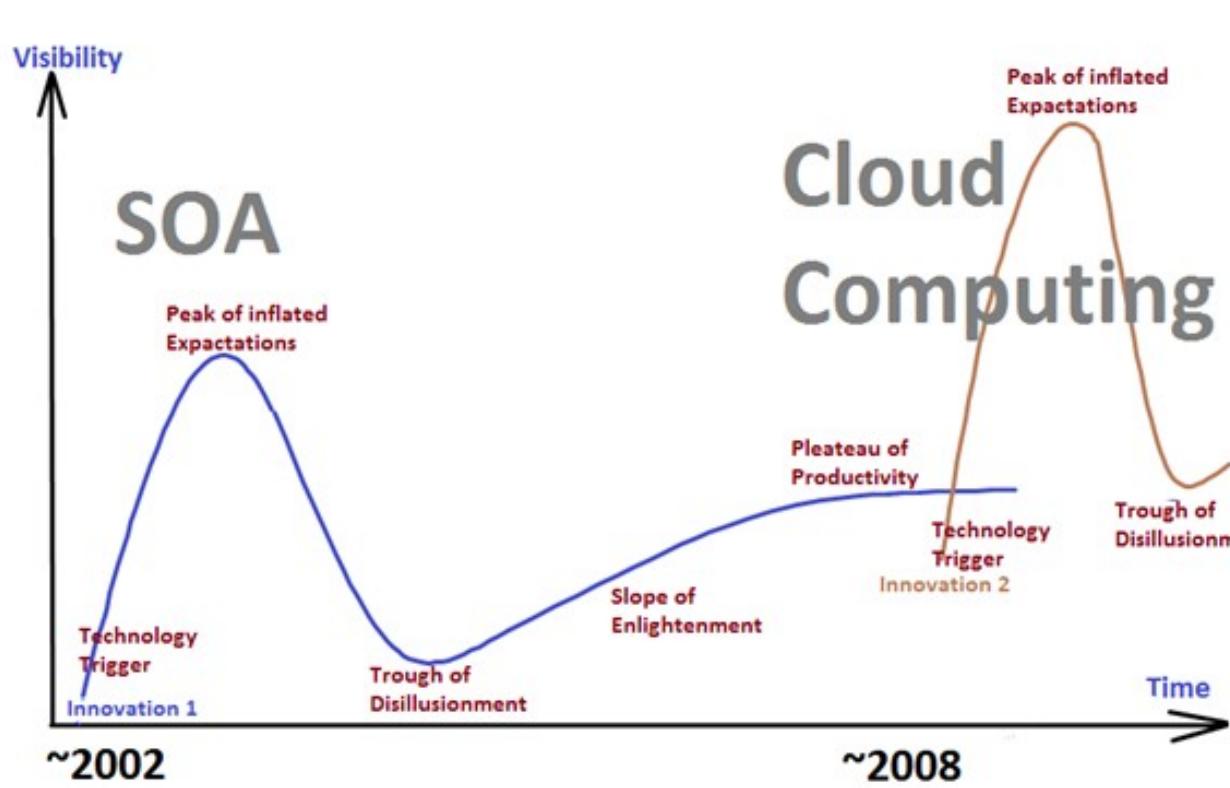


→ Tenant-configured applications

Is SOA dead?

Disclaimer

- Distributed computing is a fast evolving world
- And very much sensitive to buzzwords



Why still SOAs?

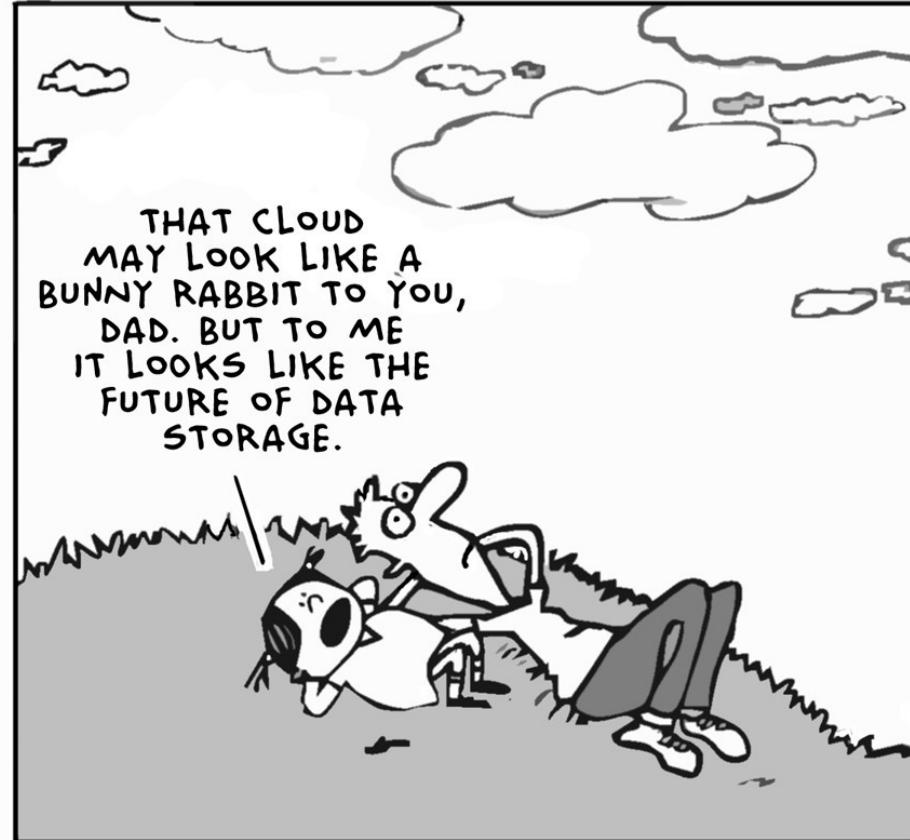
- Buzzword is fading out, the principle is not.
- Implementing technologies are very important nowadays
- SOA is implicitly used in clouds



cloud computing

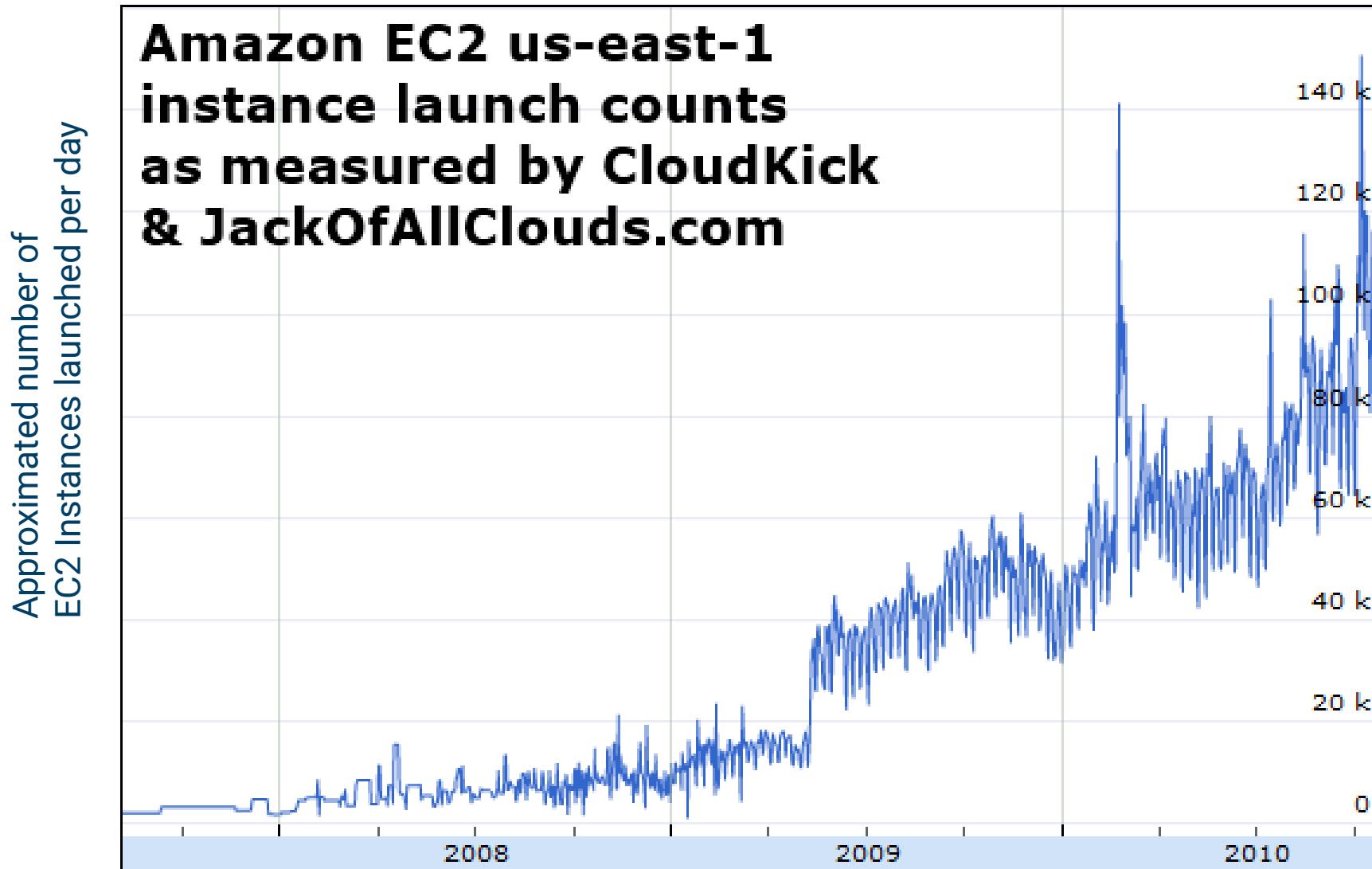
Break

See you in 15 mins. ...



Introduction to cloud computing

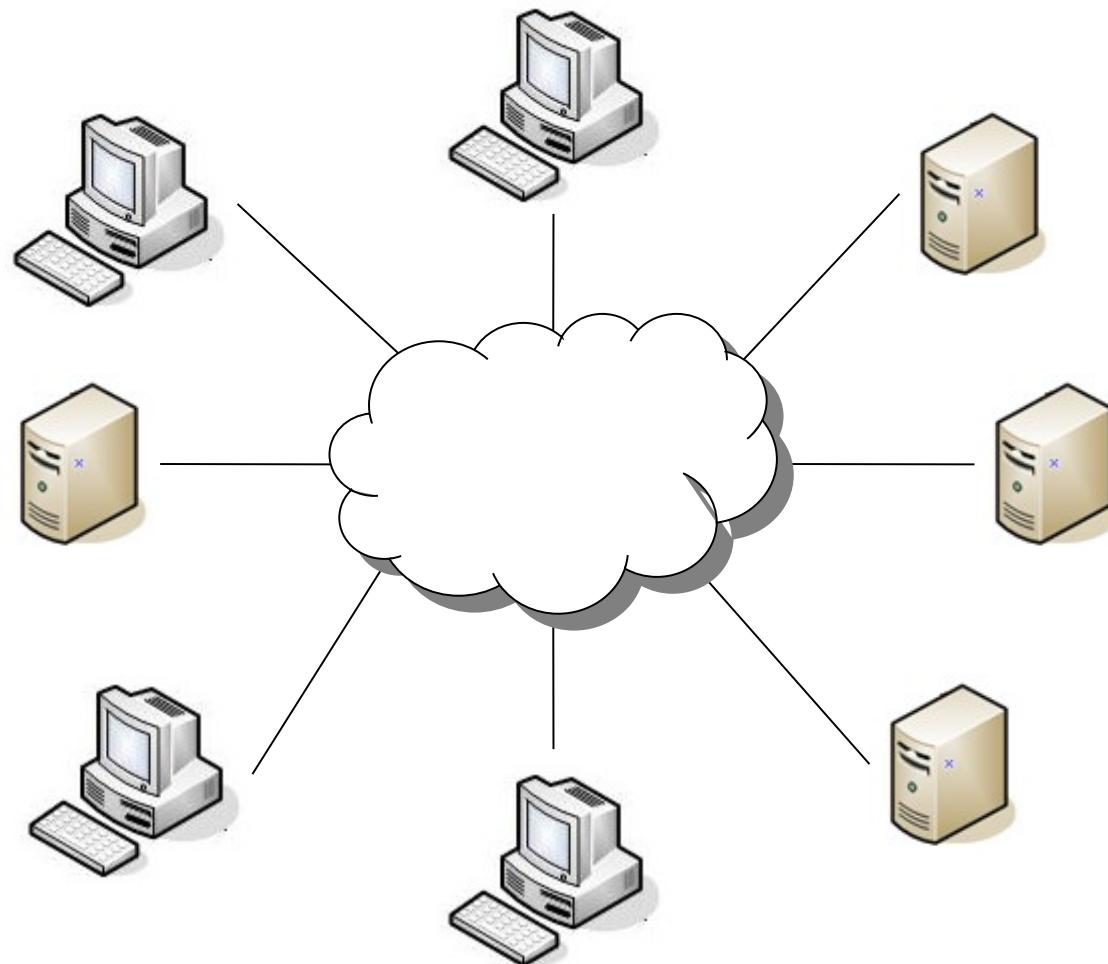
EC2 activity growth



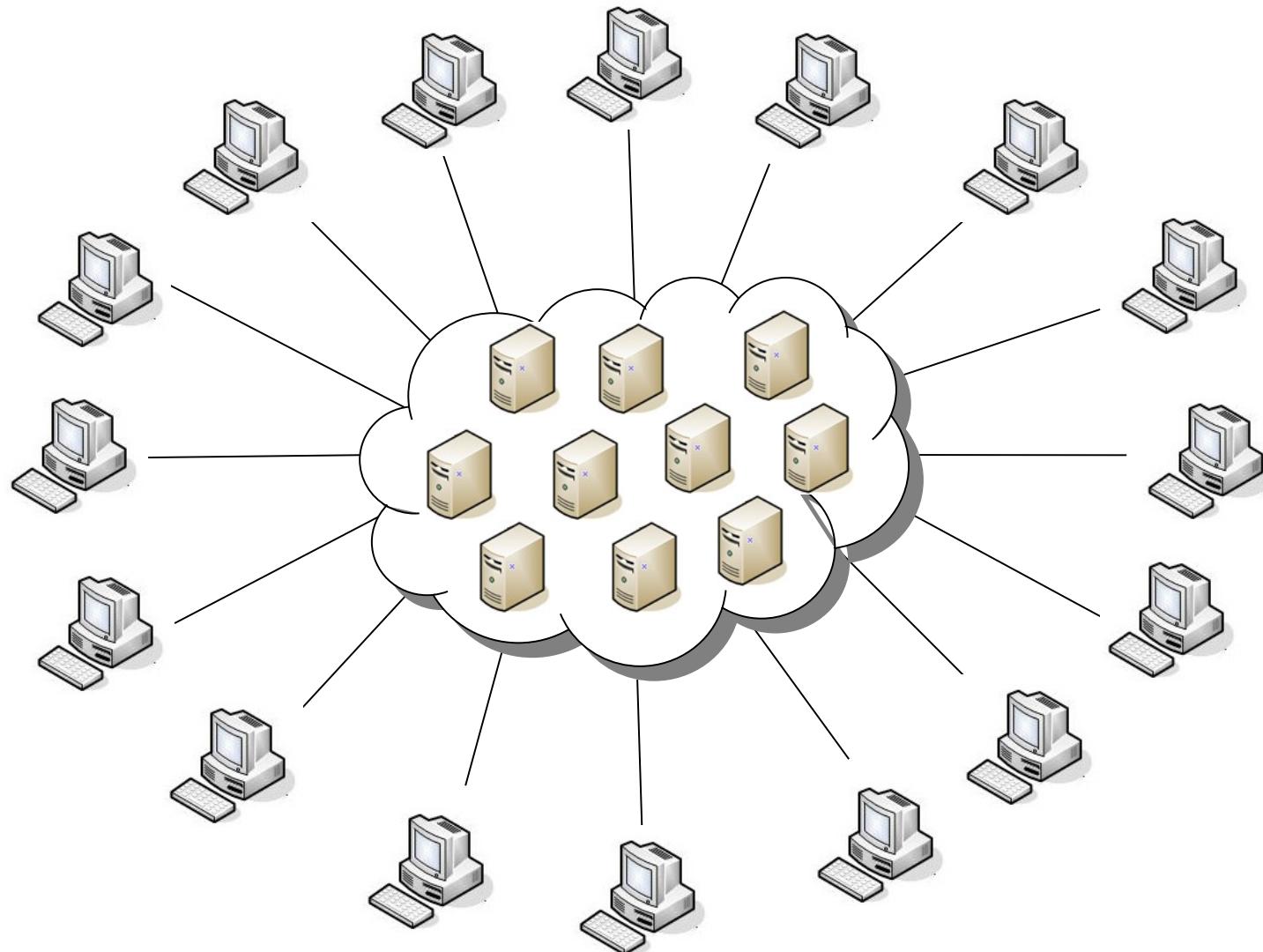
Defining a cloud...



Etymology



Etymology



Cloud computing definition

*“Cloud computing is a model for **enabling convenient, on-demand network access to a shared pool of configurable computing resources** (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”*

P. Mell and T. Grace, The NIST Definition of Cloud Computing, revision 15, 07-10-09



Present-day reality



IT services as a utility



“ If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility... The computer utility could become the basis of a new and important industry. ”

—John McCarthy, MIT Centennial in 1961

On-demand access to resources

Elasticity

Pay-per-use

D. Parkhill, The challenge of the Computer Utility. Addison Wesley Educational Publishers, 1966

Key differences

Traditional in-house IT



X as a Service



Cash up front (CAPEX)

24/7 use

Limited Capacity

Long lead time

Pay as you go (CAPEX)

Metered use

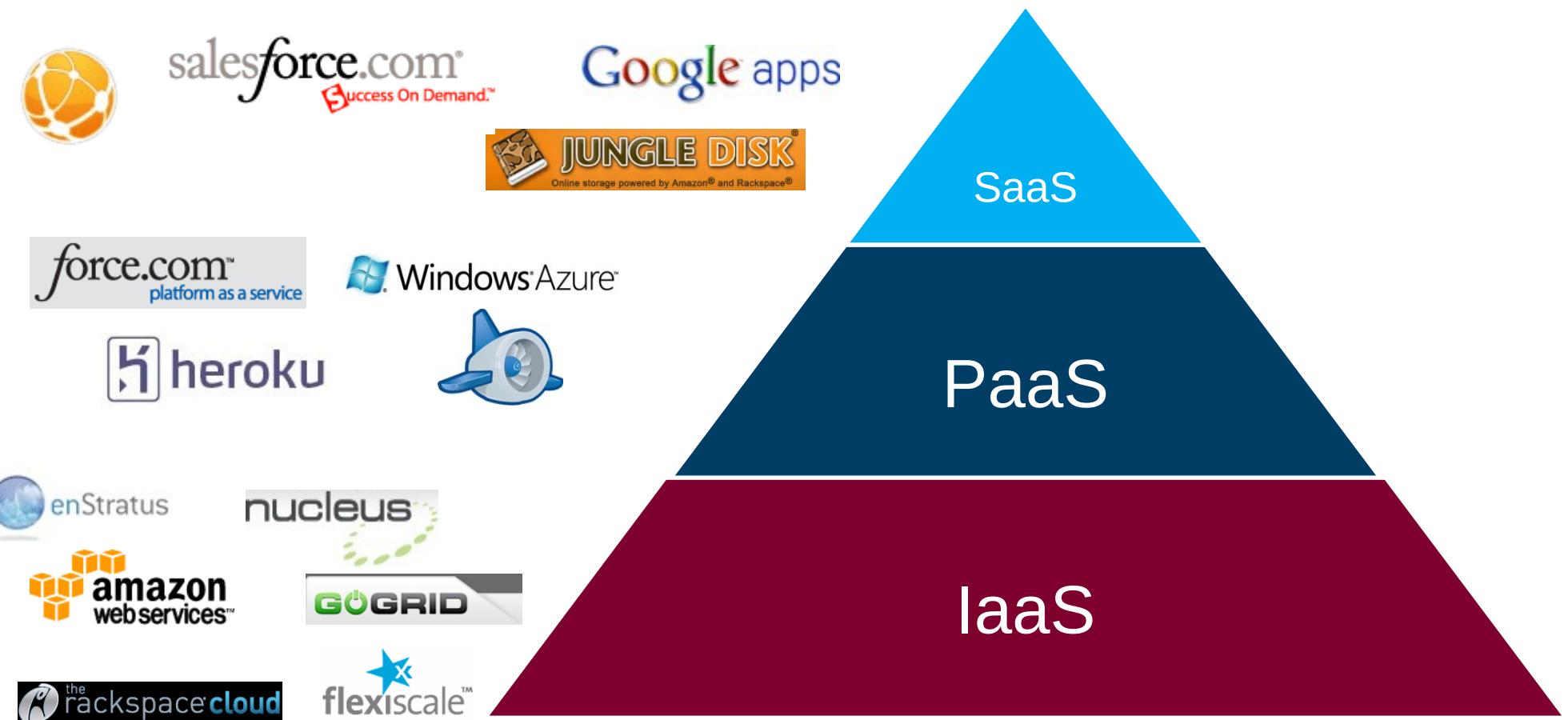
Short lead time

“Infinite” capacity

Cloud service models

...say aaS

Service models



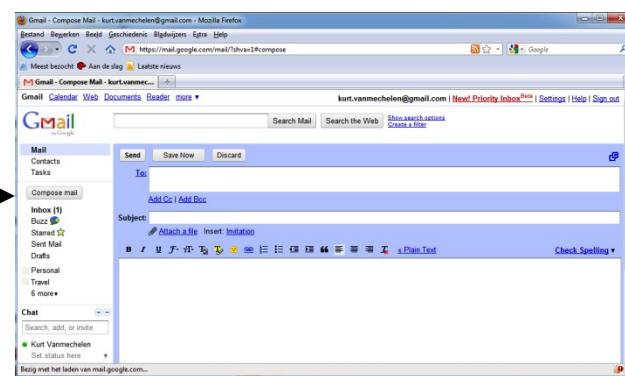
Software as a Service

*Software as a Service **delivers special-purpose software that is remotely accessible by consumers through the Internet with a usage-based pricing model.***

Use provider's application as a service

- Limited user-specific application configuration
- Accessibility through thin clients on various devices
- Automated upgrades and patching

Salesforce CRM, GMail, Google Calendar, Cloud9 BI, Google Drive, ...



Platform as a Service

*Platform as a Service **offers a high-level integrated environment to build, test and deploy custom applications.** Generally, developers will need to accept some restrictions to the type of software they can write in exchange for built-in application scalability*

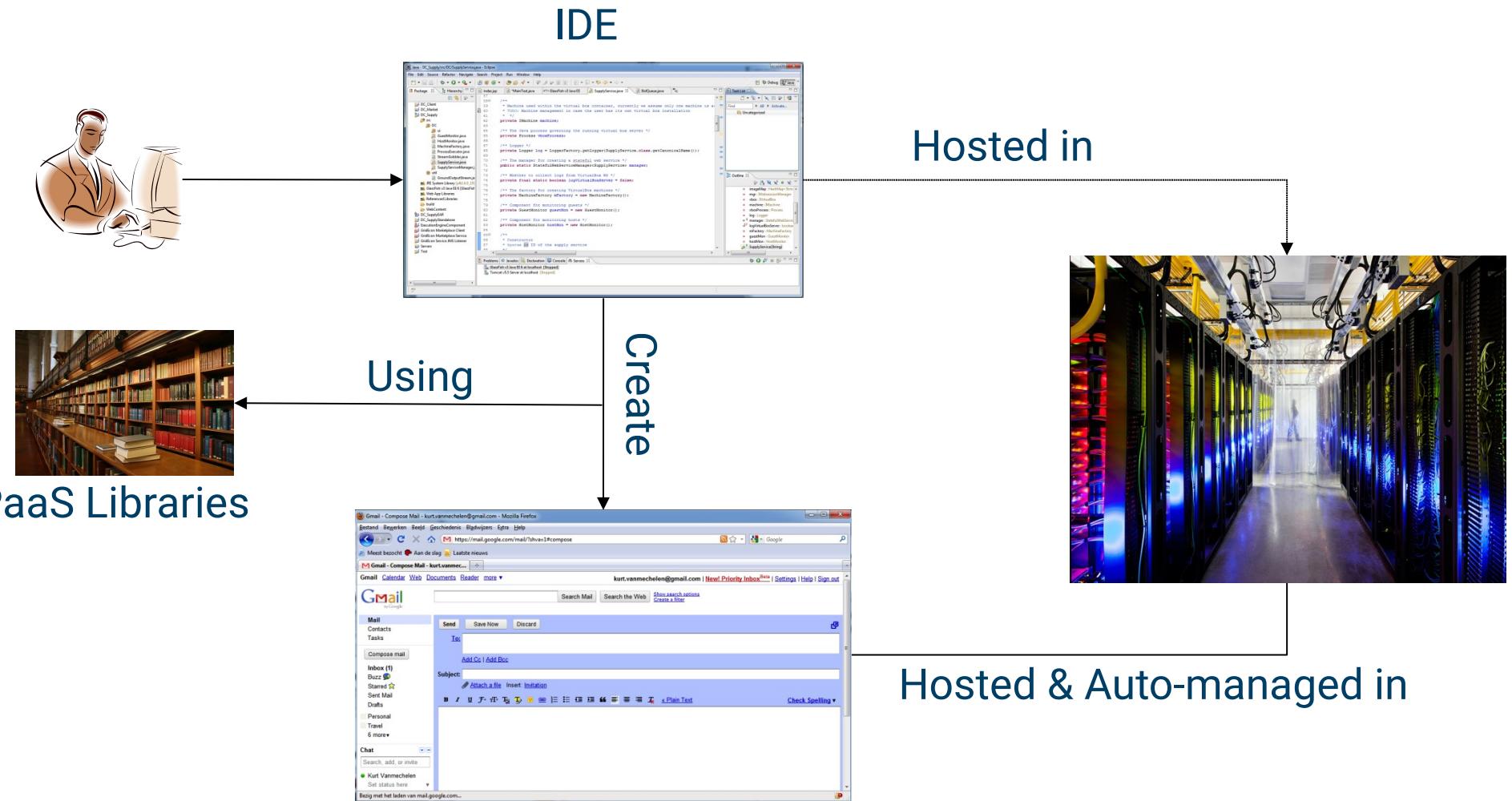
Application development environment

- Provider-constrained components and languages
 - E.g. Database access, message buses, ...
- Specific use cases (e.g. Ruby web development)
- Speedup development & deployment process
- Automated application scaling

Increased efficiency at cost of lock-in



Platform as a Service

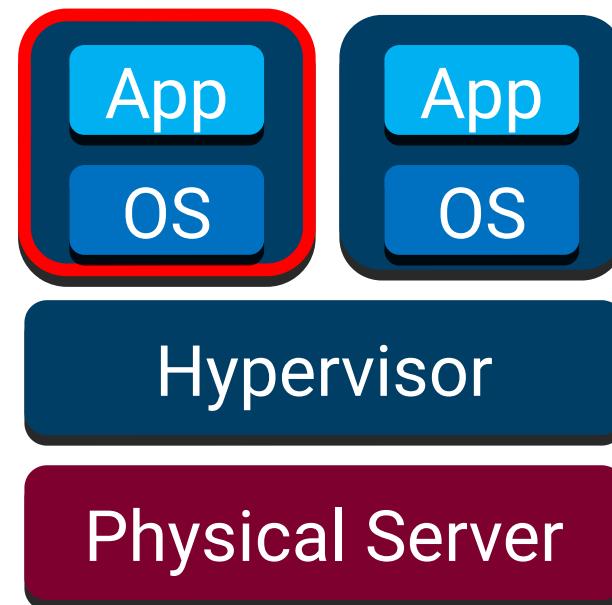


Infrastructure as a Service

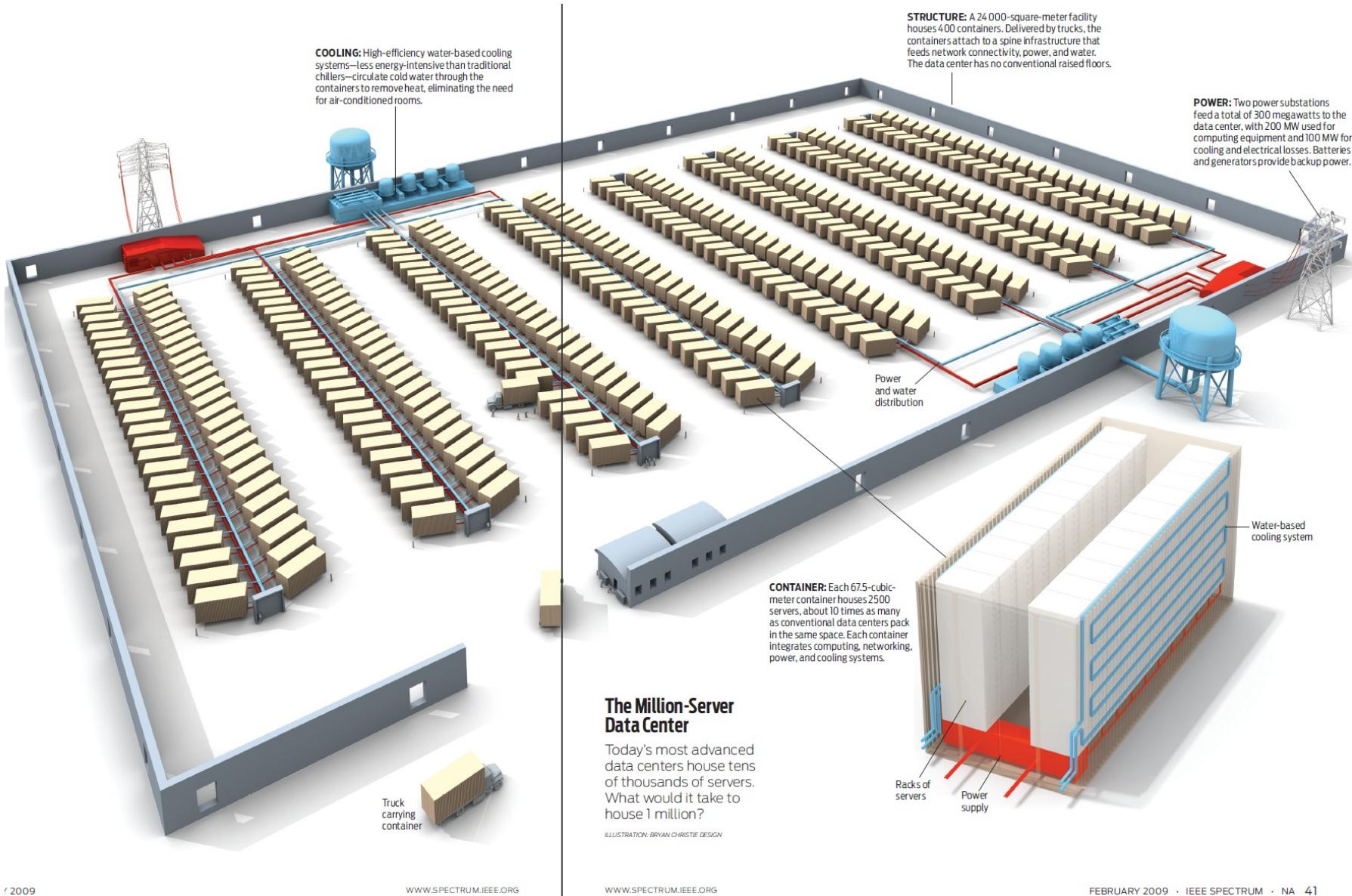
*Infrastructure as a Service **provisions hardware, software and equipment to deliver software application environments** with a resource usage-based pricing model. The resulting infrastructure can scale up and down dynamically based on application resource needs*

Provider manages underlying infrastructure (**Multi-tenancy**)

Unit of deployment : Virtual Machine (VM)



A question of scale...



A question of scale...



SUN MD S20 (280 quad-core servers, 187.5 kW, 12.6 kW/m²)
(01/2008)

A question of scale...

The Cirrascale FOREST Container

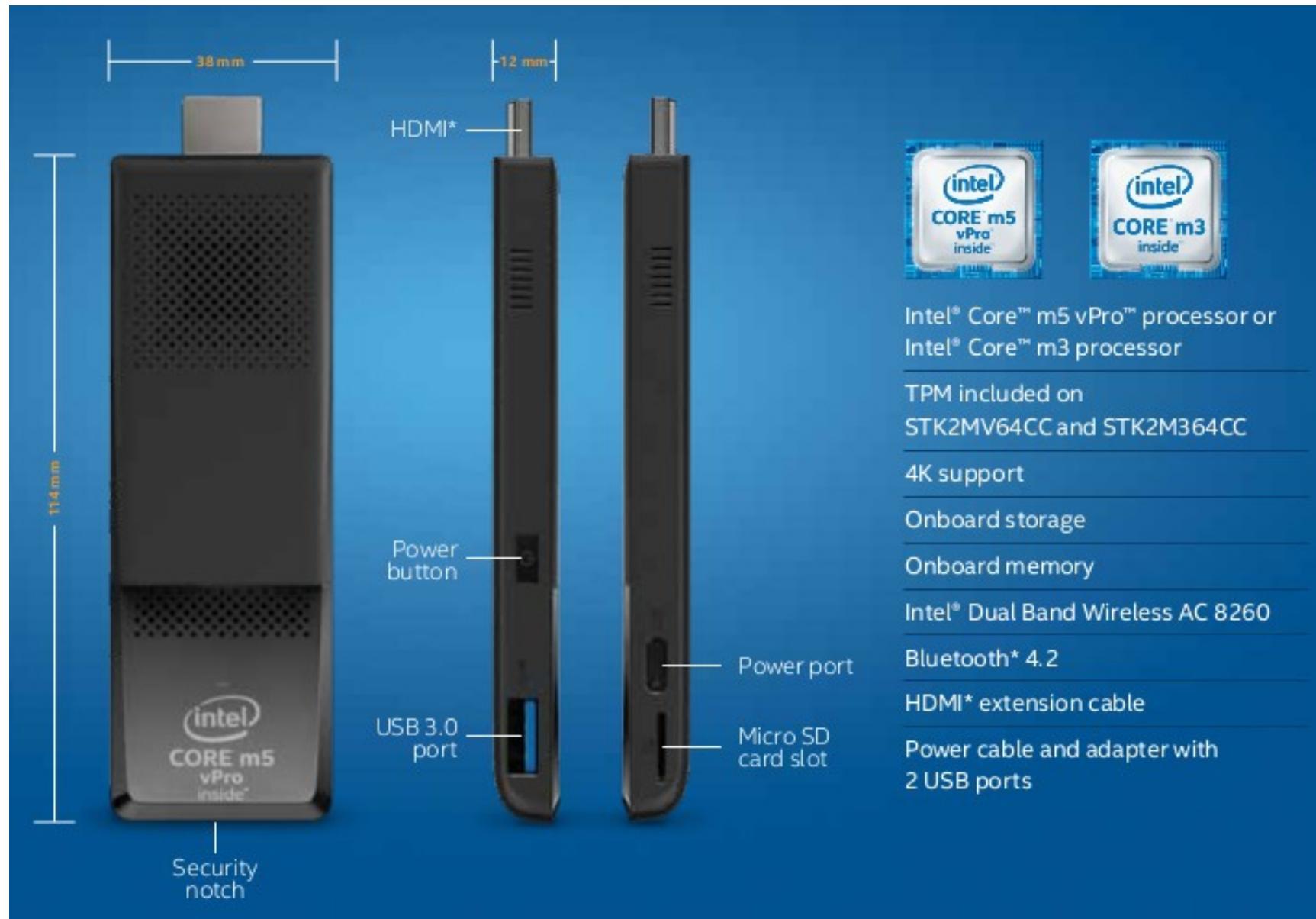
Flexible, Open, Reliable, Energy-efficient, Scalable, and Transportable



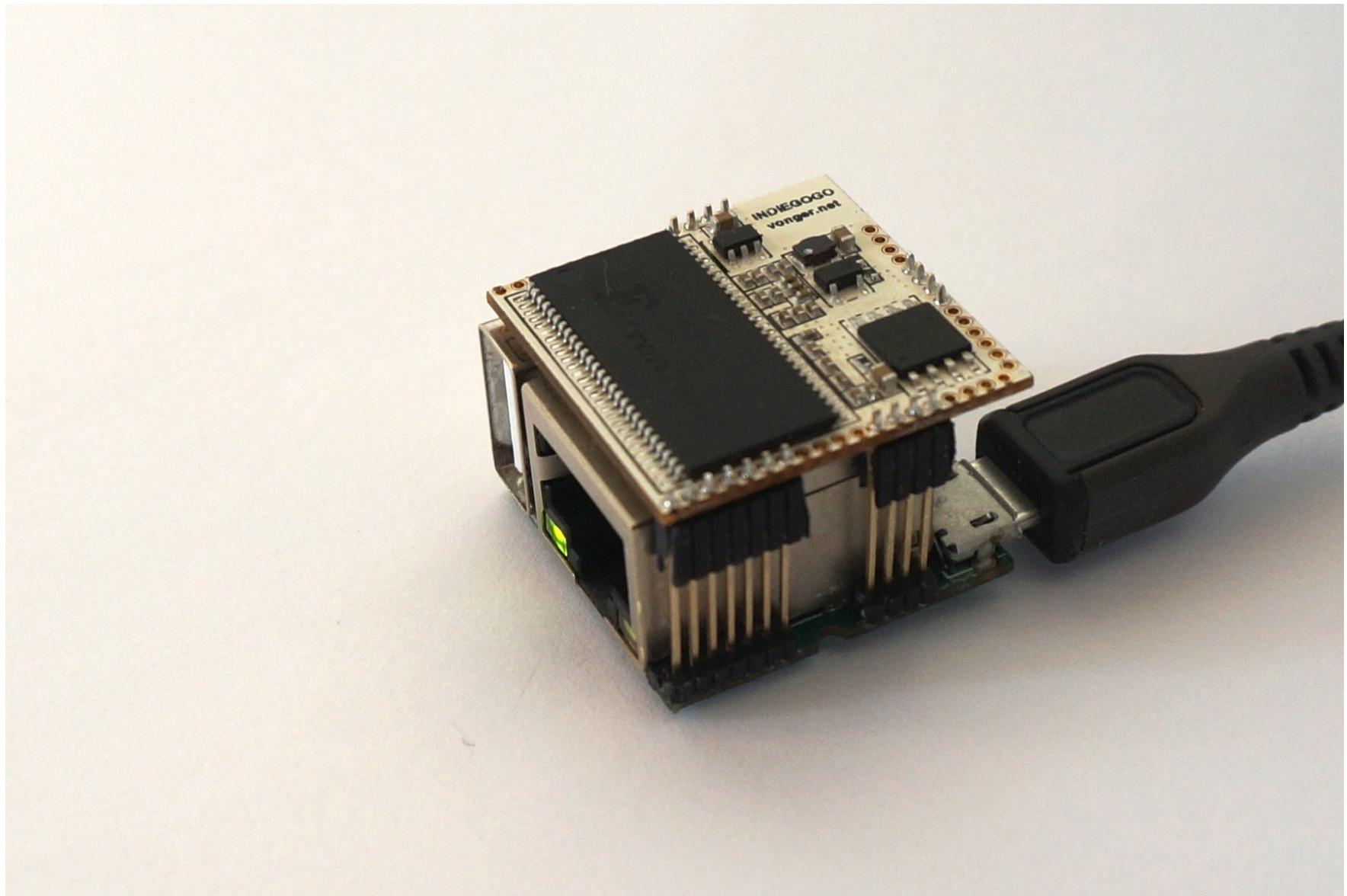
- Houses 26PB of storage or 2,880 servers
- Utilizes patented Vertical Cooling Technology
- Can use external temperatures to reduce costs
- Superior energy-efficiency, serviceability
- Can host wide-range of third-party equipment
- Cross-compatible blades with BR2-FE

23,040 cores, 350kW, (12,2m x 2,4m x 2,9m)

Or a small scale...



Or a small scale...

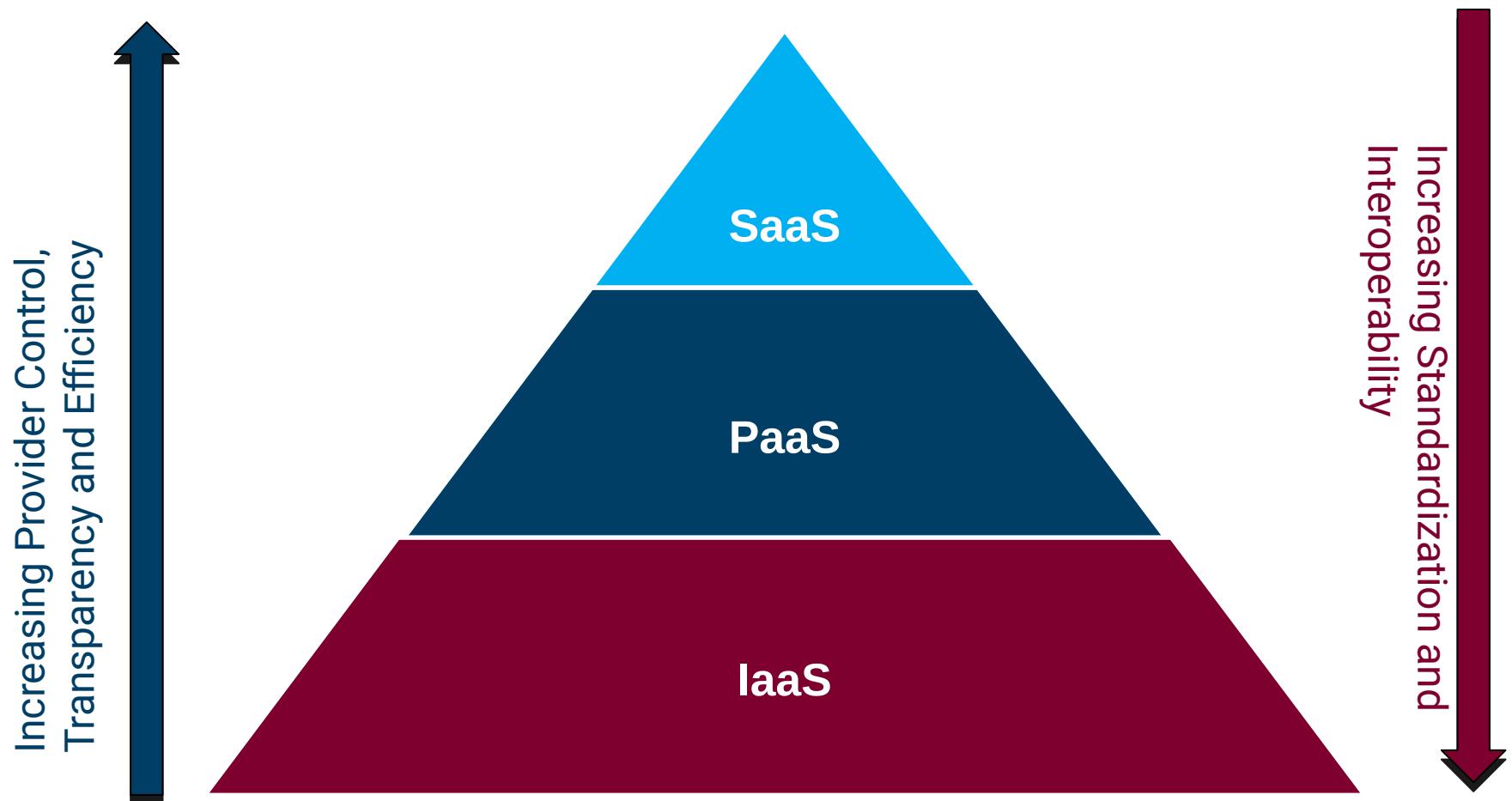


OpenWRT \$12 Tiny Gateway

Service models

These models are not orthogonal!

SaaS can be build on top of PaaS, which can be build on top of IaaS



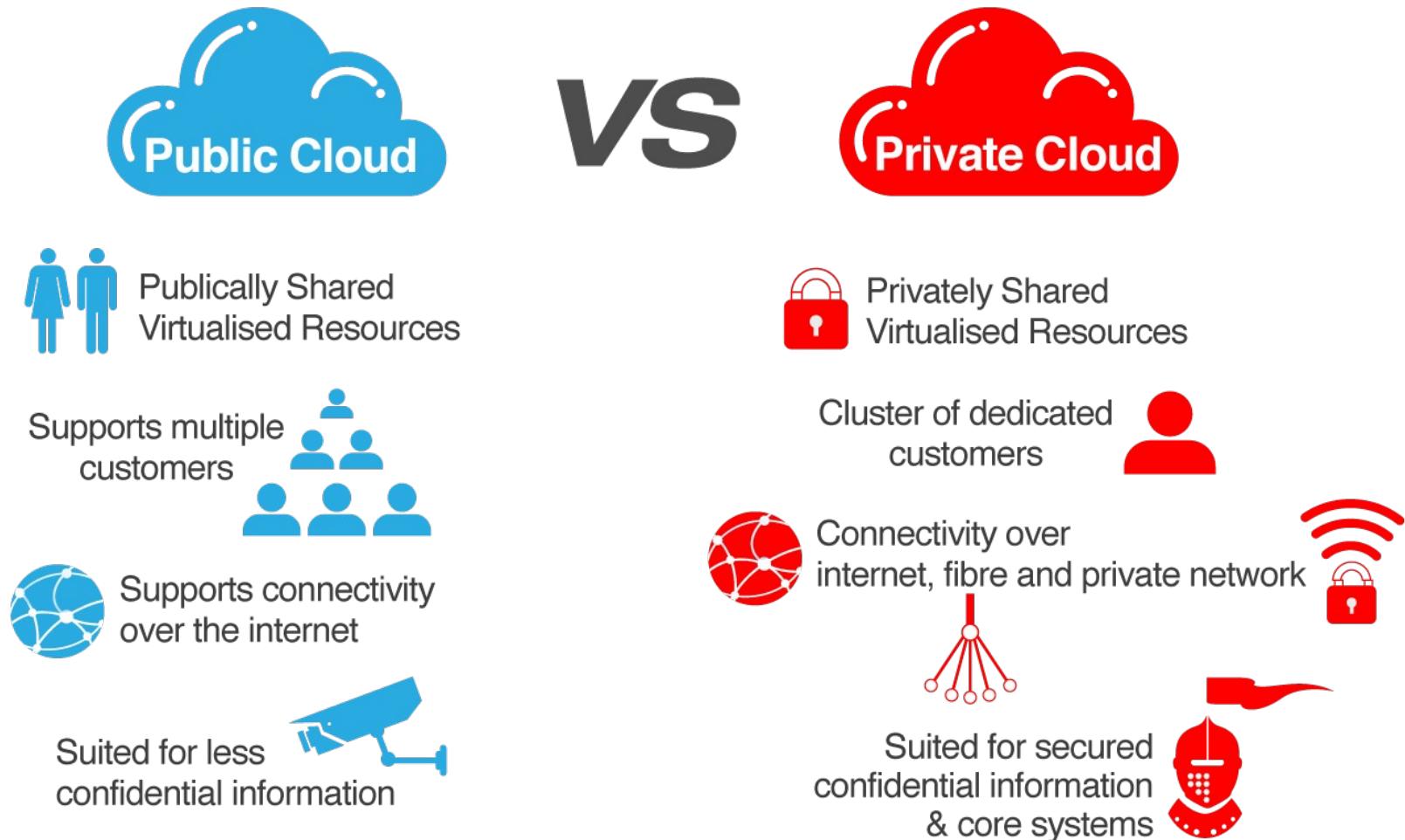
The 3 main enablers

Virtualization Technology & Standards

Web Service Technology & Standards

Large capital investments by major players & economies of scale

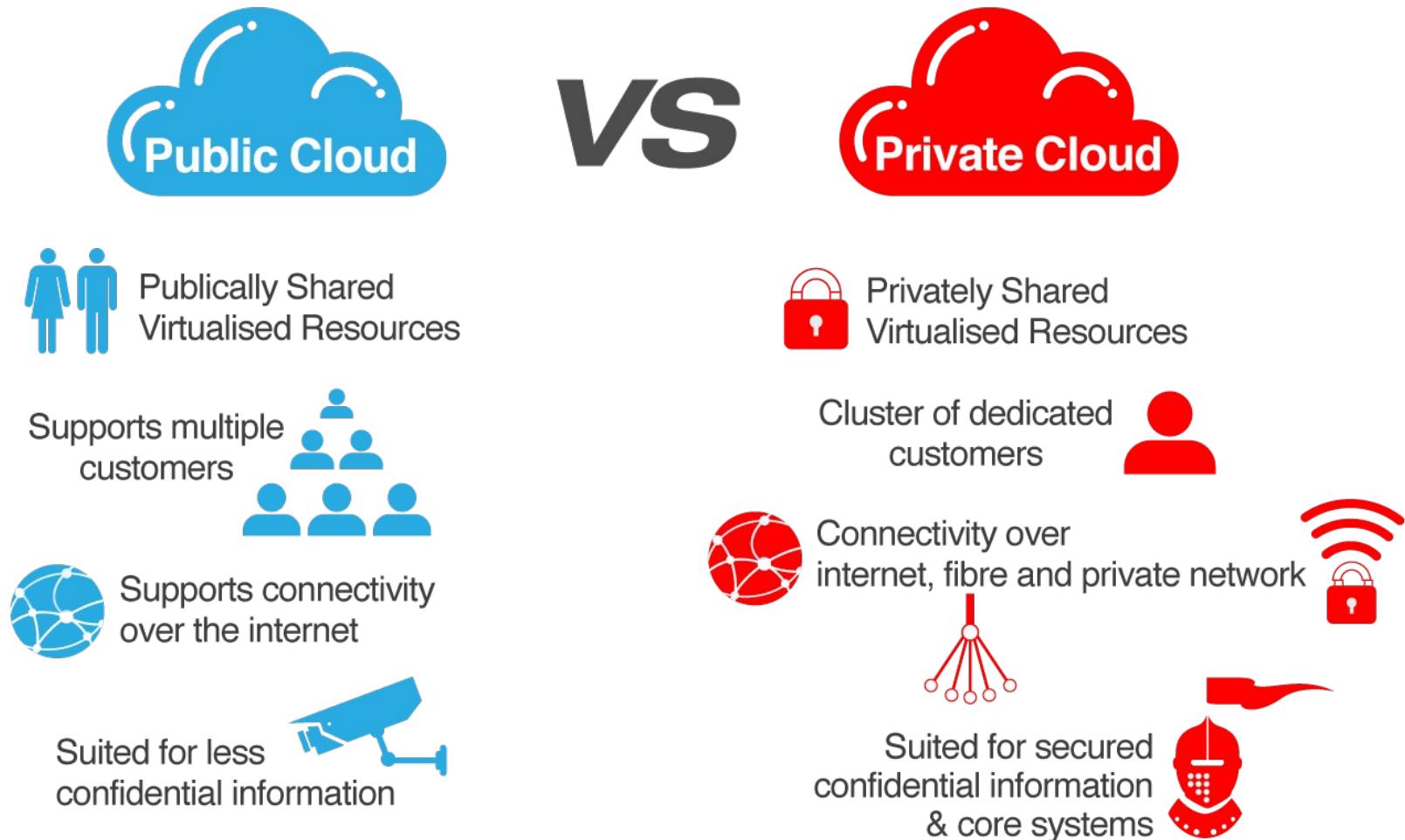
Deployment models



Other deployment models

- Community clouds: hosted by several organizations
- Hybrid clouds: combinations of above

Deployment models



Other deployment models

- Community clouds: hosted by several organizations
- Hybrid clouds: combinations of above

Payment models



1. Per-Instance billing:

- Pay for every hour a VM is used
- You pay for idle machines



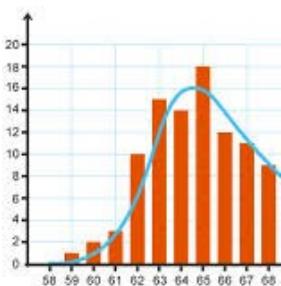
2. Reserved usage:

- Up-front payment & reservation
- Lower rates



3. Bidding:

- You define a maximum instance price
- Price varies on the load of the cloud

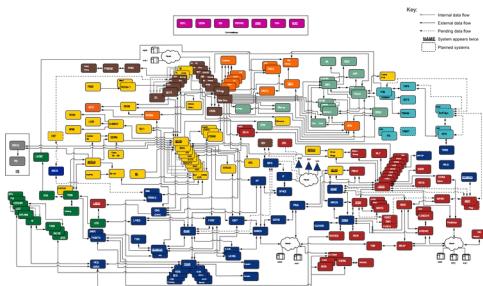


4. Actual usage:

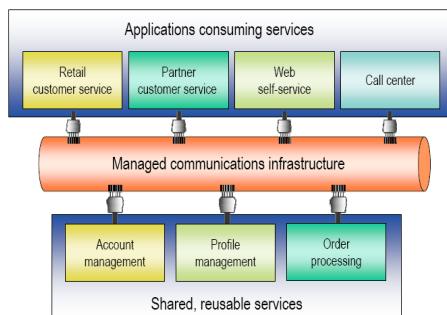
- Mostly in PaaS models

Wrap-up

Quick run-down of decades of distributed computing history



Tightly coupled distributed systems
e.g., in telecom



Service Oriented Architectures
Term is dead, but idea is not



Introduction to cloud computing
X as a Service

Questions?

Further Reading

- **Service Oriented Architectures**

https://resources.sei.cmu.edu/asset_files/Webinar/2008_018_101_47057.pdf

- **Multi-tenancy**

<https://www.sitepoint.com/multitenancy-and-google-app-engine-gae-java/>

<https://www.confluent.io/blog/sharing-is-caring-multi-tenancy-in-distributed-data-systems/>

- **CORBA**

<https://www.ibm.com/docs/en/integration-bus/9.0.0?topic=corba-common-object-request-broker-architecture>

- **Hypervisor in Cloud Computing**

<https://www.educba.com/hypervisor-in-cloud-computing/>

- **Relevant Cloud Computing statistics of 2022**

<https://www.cloudwards.net/cloud-computing-statistics/Web services>

Service Oriented Architectures (SOAs)