Distributed Systems – Current Trends in Distributed Systems

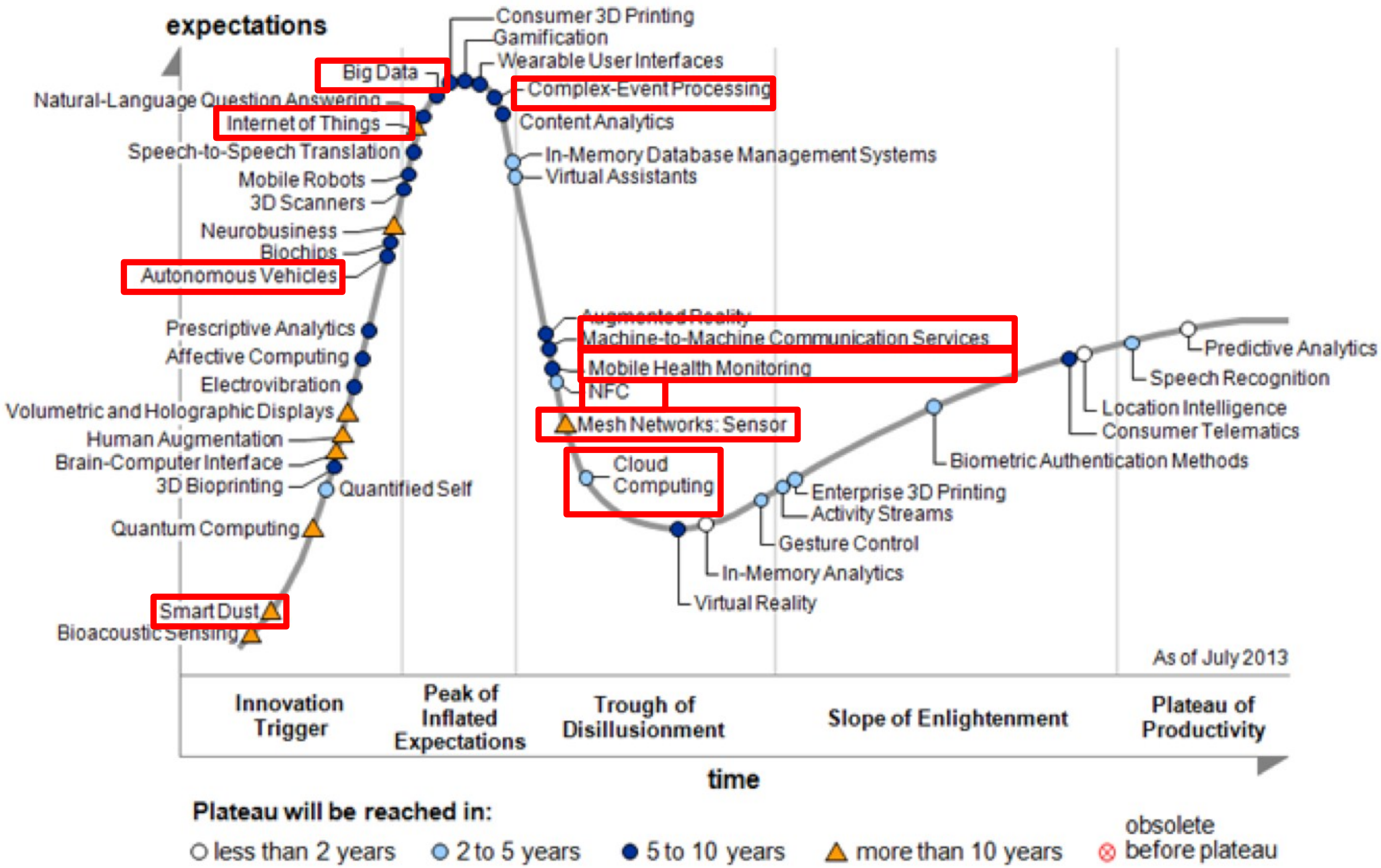
Prof. Dr. Schahram Dustdar
Distributed Systems Group
Vienna University of Technology

dustdar@infosys.tuwien.ac.at

Outline

1. Overview
2. Peer-to-Peer Computing
3. Service-oriented Computing
4. Cloud Computing
5. Epilogue

Gartner Hype Cycle for Emerging Technologies, 2013



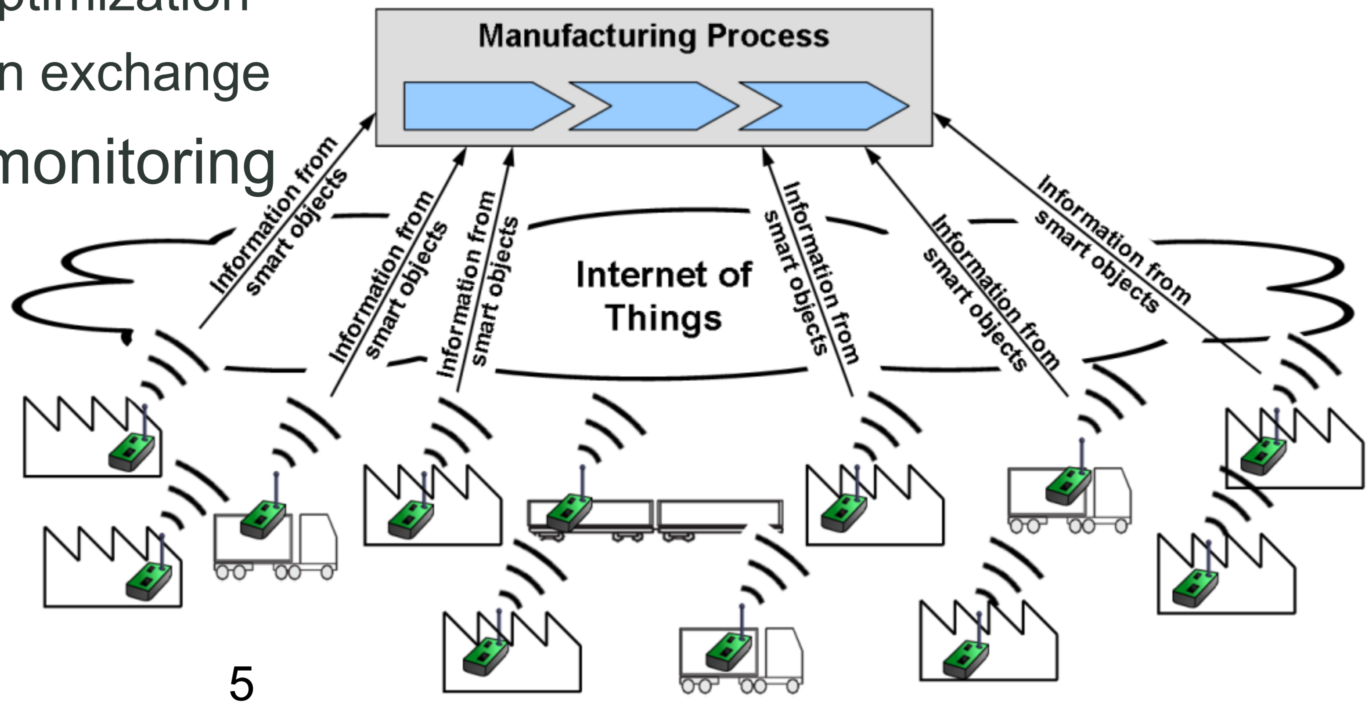
Distributed Systems Group

Major Trends in Distributed Systems I

- Internet of Things (IoT):
  - Physical objects are seamlessly integrated into the information network
  - Physical objects become active participants in business processes
  - Physical objects become “Smart Objects”
  - Technologies: RFID, sensor networks, Internet Protocol version 6 (IPv6)

IoT – Example: Factories of the Future

- Combining the power of independent factories
- Achieving complex manufacturing processes
- Providing concrete tools for
  - Process creation
  - Process optimization
  - Information exchange
- Real-time monitoring



Major Trends in Distributed Systems II

- Internet of Services (IoS):
  - Software services are provided through the Internet
  - Technologies: REST, WSDL, SOAP
  - Foundation for Cloud Computing

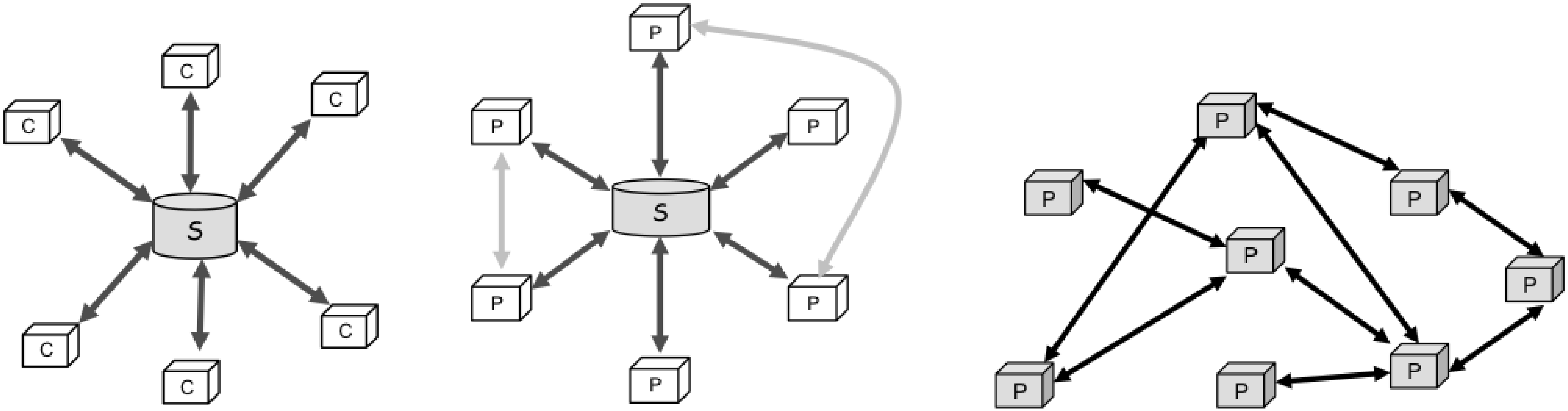
- Service-oriented Architectures vs. IoS:
  - IoS = Global SOA?
  - SOA: Originally mainly a concept to organize IT software architectures in an organization

Outline

1. Overview
2. Peer-to-Peer Computing
3. Service-oriented Comp
4. Cloud Computing
5. Epilogue

Slides are based on the book "Peer-to-Peer Systems and Applications", LNCS Vol. 3485, Springer and lecture "Peer-to-Peer Systems and Applications" (TU Darmstadt)

Peer-to-Peer: Overview



(a) Client/Server

(b) Hybrid

(c) Peer-to-Peer

Peer-to-Peer

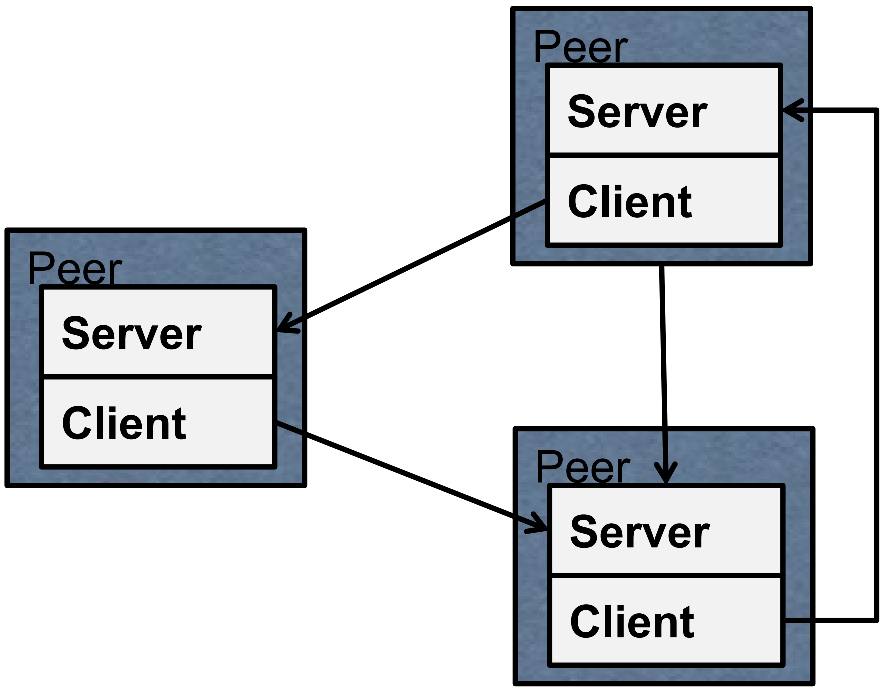
- Components directly interact as peers by exchanging services
- Request/reply interaction without the asymmetry found in the client-server pattern – all peers are equal
- Each peer component provides and consumes similar services

What is P2P?

- Definition according to Oram et al.:
  - A Peer-to-Peer (P2P) system is „a self-organizing system of equal, autonomous entities (peers) [which] aims for the shared usage of distributed resources in a networked environment avoiding central services.“
  - „A system with completely decentralized self-organization and resource usage.“
- Derived key characteristics of a P2P system:
  - Equality – All peers are equal (peer = gleichgestellt)
  - Autonomy – No central control
  - Decentralization – No centralized services
  - Self-organization – No coordination from outside
  - Shared resources – Peers may use resources provided by other peers

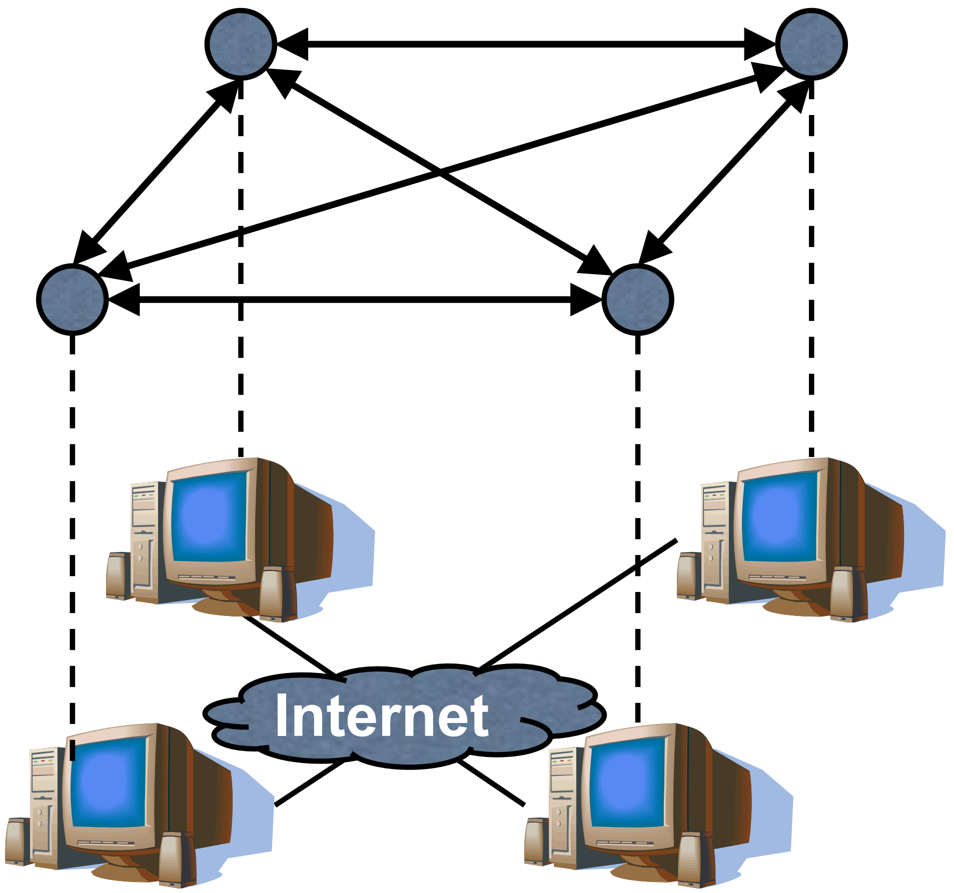
Peers

- Peers
  - Are nodes running in some P2P overlay
  - Have all the same capabilities (ability to act in any role)
  - Can act as “clients” and “servers” at the same time



Overlay-Network

- Composed of direct connections between peers
- Typically an “overlay“ network on top of a network (e.g., the Internet)
- But completely independent from physical network, due to abstraction of the TCP/IP layer
- Separate addressing scheme

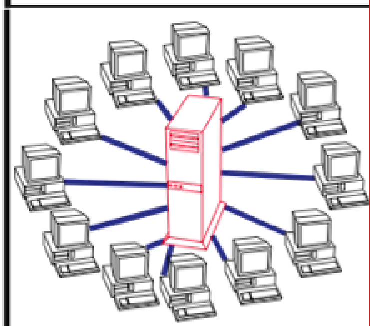
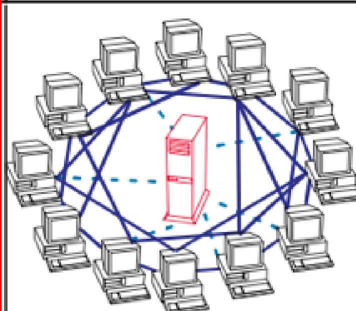
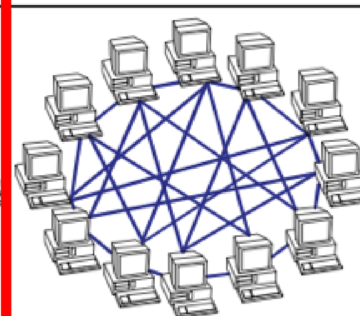
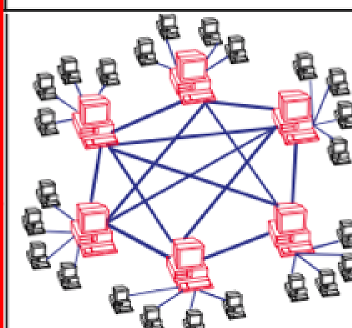
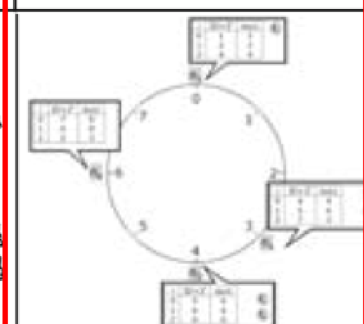


P2P: Application Areas

- Several application areas:
  - VoIP (Skype/FastTrack)
  - Media streaming (Joost)
- In 2006, P2P made up 70% of the Internet traffic (CacheLogic Research):
  - P2P accounts for ~19% of fixed access traffic in North America according to Sandvine (2010/11)
  - Bittorrent is the single biggest application regarding upstream traffic in North America in 2010/11 (52%)
- Obviously, File Sharing is one area where P2P is heavily applied:
  - Napster (1st Generation Centralized P2P)
  - Gnutella 0.4 (1st Generation Pure P2P)
  - Gnutella 0.6, FastTrack/KaZaA (2nd Generation Hybrid P2P)
  - Kademlia (foundation for trackerless BitTorrent and eDonkey) → Structured P2P

Reasons for Application of P2P

- Costs: Computing/Storage can be outsourced (this is the major reason why Skype applies P2P)
- High Extensibility (easy to add further resources)
- High Scalability (system can grow to a very large number of peers)
- Fault Tolerance: If one peer fails, the overall system will nevertheless work
- Resistance to lawsuits…

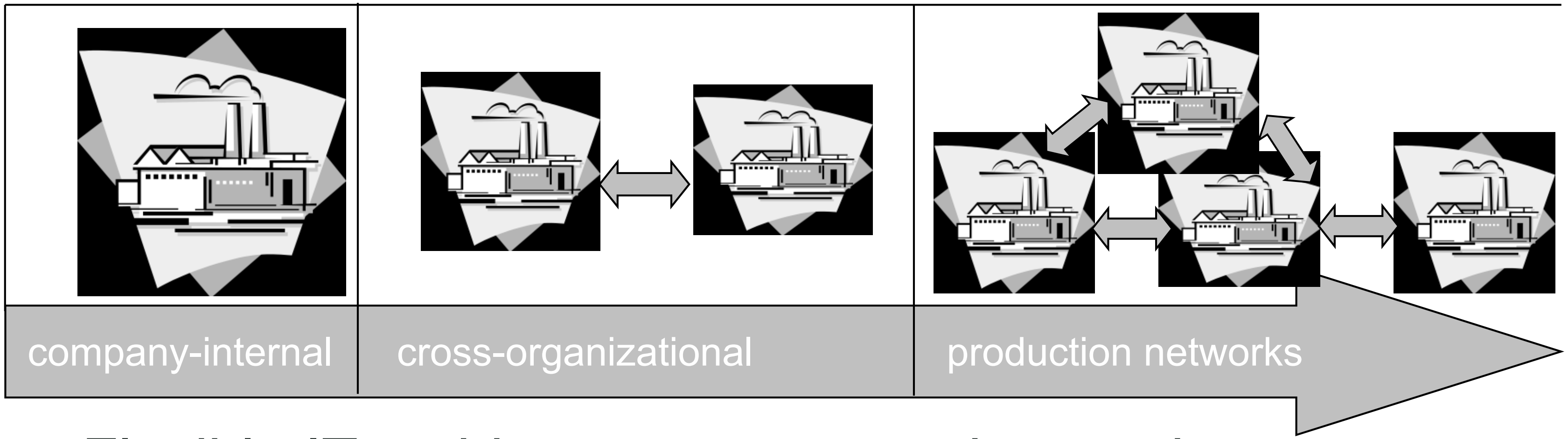
Client-Server	Peer-to-Peer			
	1. Resources are shared between the peers 2. Resources can be accessed directly from other peers 3. Peer is provider and requestor (Servent concept)			
	Unstructured P2P			**Structured P2P**
	1st Generation		**2nd Generation**	
	Centralized P2P	*Pure P2P*	*Hybrid P2P*	*DHT-Based*
1. Server is the central entity and only provider of service and content. → Network managed by the Server 2. Server as the higher performance system. 3. Clients as the lower performance system Example: WWW	1. All features of Peer-to-Peer included 2. Central entity is necessary to provide the service 3. Central entity is some kind of index/group database Example: Napster	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → No central entities Examples: Gnutella 0.4, Freenet	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → dynamic central entities Example: Gnutella 0.6, JXTA	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → No central entities 4. Connections in the overlay are "fixed" Examples: Chord, CAN
				

Outline

1. Overview
2. Peer-to-Peer Computing
3. Service-oriented Computing
4. Cloud Computing
5. Epilogue

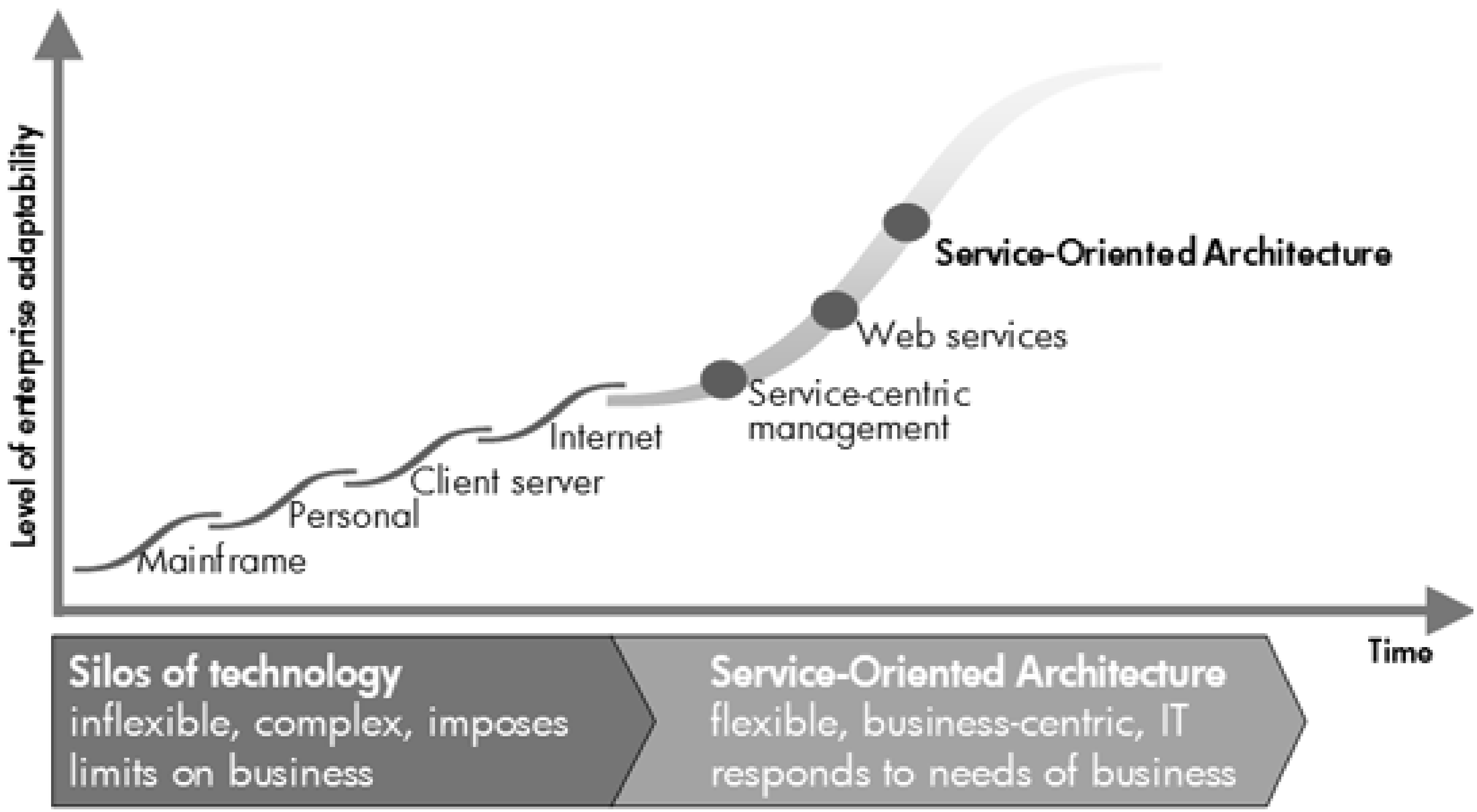
Motivation

- Major Trend since the 1990s:
  - Globalization, deregulation of markets
  - Cross-organizational workflows and business processes are of major importance
  - Business Process Outsourcing (BPO)
  - Flexibility of business processes is a key success factor



- Flexible IT architectures are a major requirement:
  - Integration of legacy systems
  - Coupling to IT systems of business partners

Motivation – A Shift of Paradigms

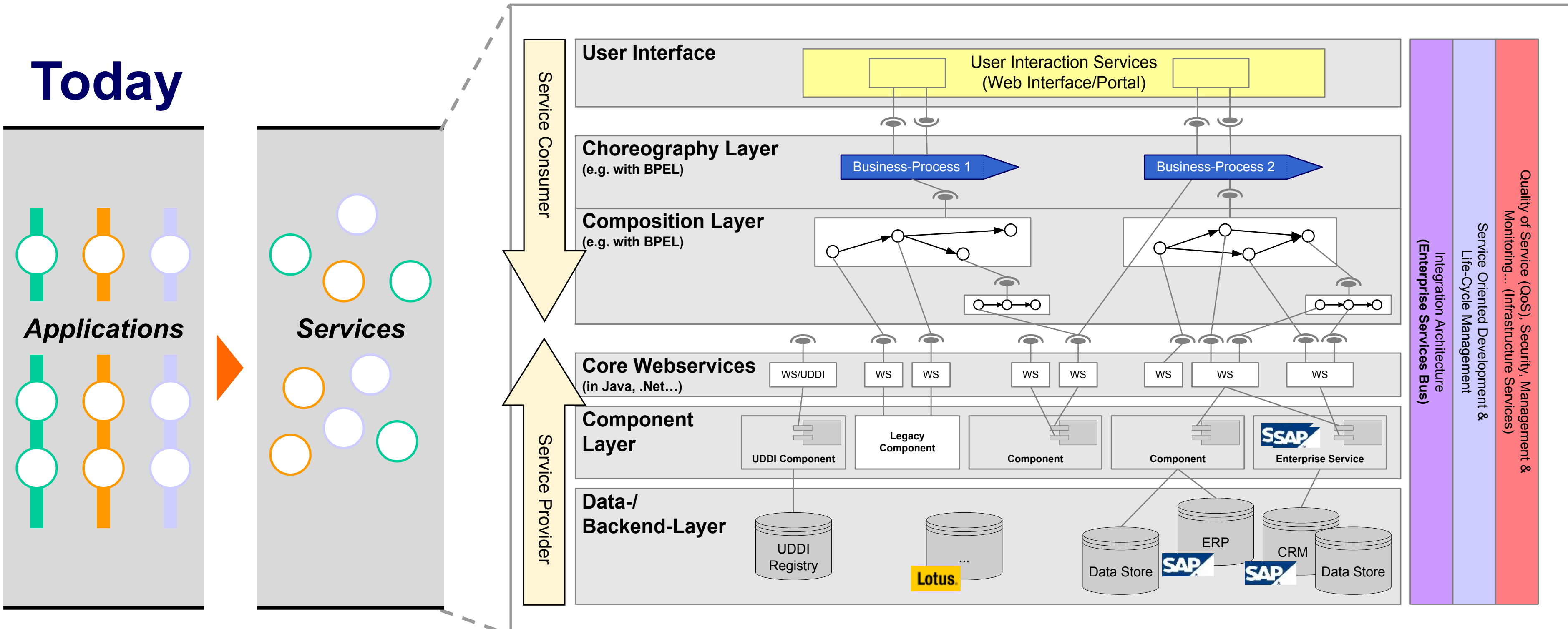


Vision of a Service-oriented Architecture

"Loosely Coupled, Process Driven Services and Components"

Tomorrow

Today



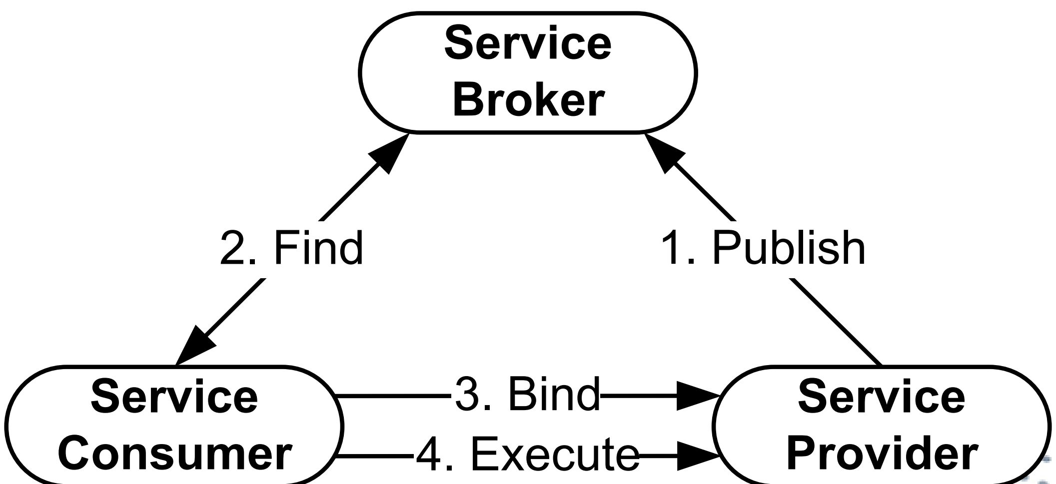
Discrete Applications

Services

Source: IBM 2007

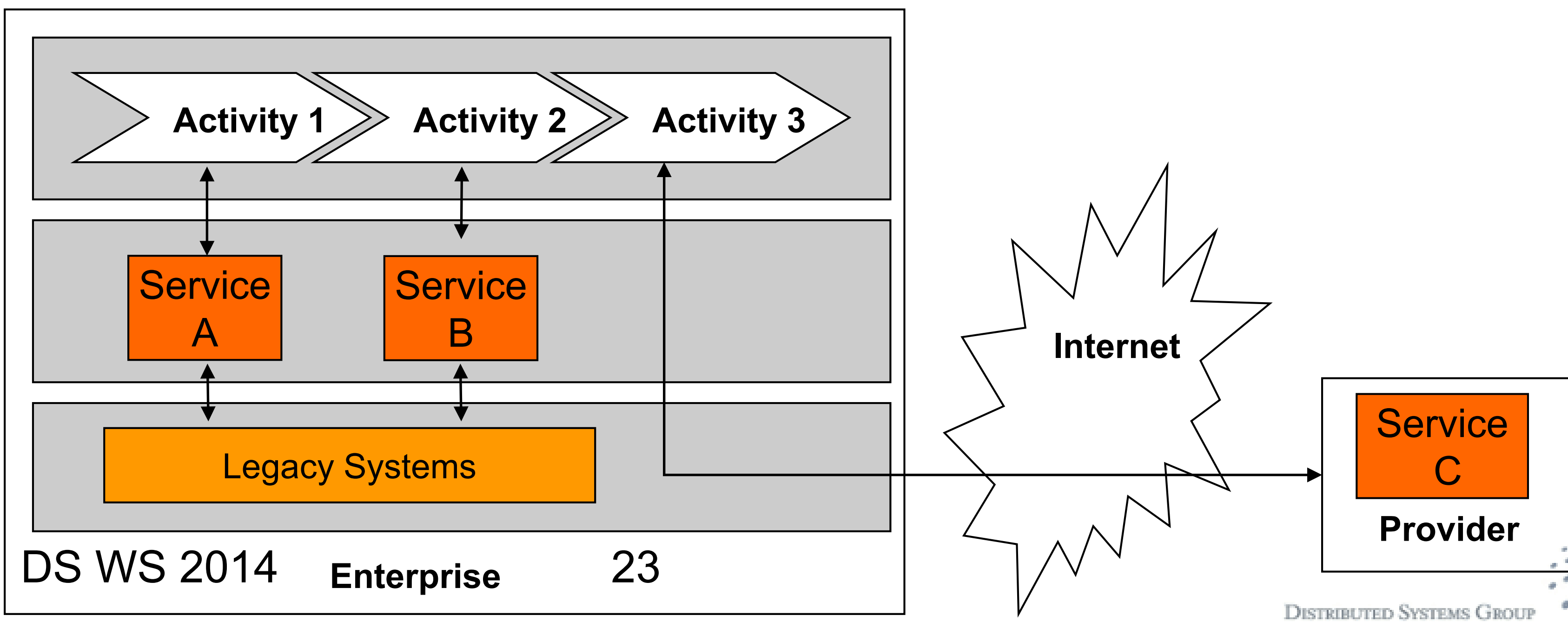
SOA – Overview and Roles

- Service-oriented Architectures:
  - IT architecture made up from single services, i.e., self-contained software components with a distinct functionality
  - Complex applications arise from the coupling of single services, e.g.,
    - Service-based workflows
    - Mashups
  - However, it is also possible to invoke single services
- Roles in a Service-oriented Architecture
  - Service Provider
  - Service Consumer
  - Intermediary (optional), e.g., Service Broker



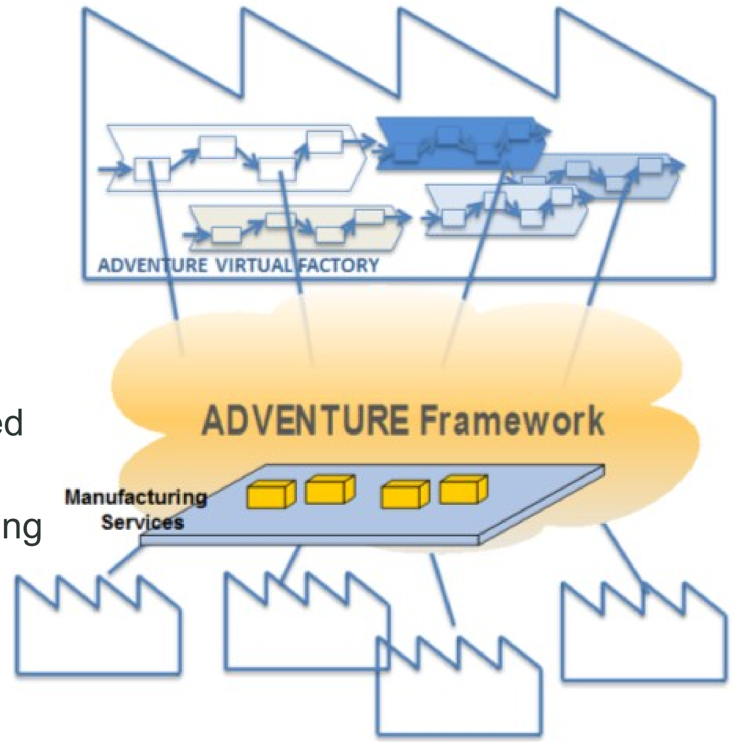
Workflows and Services

- Workflows and Services:
  - Workflows are IT-enabled business processes
  - Services can be composed to workflows (2-level-programming)
  - Services wrap functionality of legacy systems (e.g. Service A/B)
  - Integration external services (e.g. Service C)
- Services support rapid composition of distributed workflows



Example for IoT and IoS: ADVENTURE – The Plug-and-Play Virtual Factory

- Virtual Factory
  - Multiple factories may form a virtual factory
  - Integrated ICT
    - Leverage information exchange
  - Interoperability at a deeper technical level
    - Ensuring that factories can be technically connected
- Plug
  - Factories provide information
    - Semantically enriched descriptions of offered manufacturing capabilities and products
    - Sensor technologies to monitor manufacturing processes
- Play
  - Factories model manufacturing process
    - Manufacturing processes modeled as composition of services
  - Identify particular partners who offer a distinct product



ADVENTURE – The Plug-and-Play Virtual Factory

Distributed Systems Group

Outline

1. Overview
2. Peer-to-Peer Computing
3. Service-oriented Computing
4. Cloud Computing
5. Epilogue

Slides are based on “A View of Cloud Computing”, Armbrust et al.,Communications of the ACM, Vol. 53, No. 4, April 2010 and
The NIST Definition of Cloud Computing

Motivation – Want milk?

- Buy a cow:
  - High upfront investment
  - High maintenance cost
  - Produces a more or less fixed amount of milk
  - Stepwise (discrete) scaling

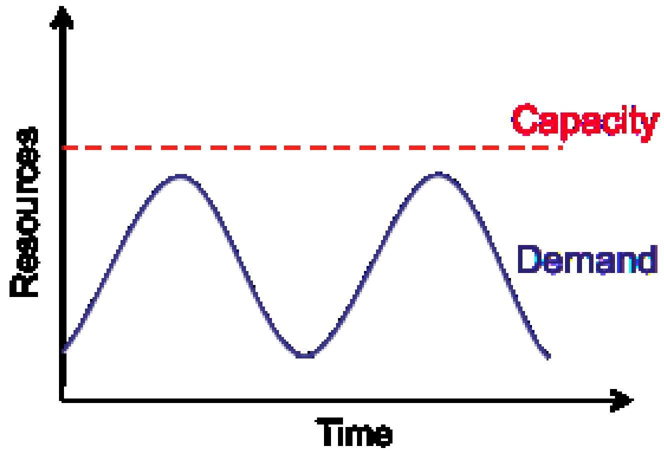
- Buy bottled milk:
  - Pay-per-use
  - Lower maintenance cost
  - Linear (continuous) scaling
  - Fault-tolerant

Use Cases for Cloud Computing

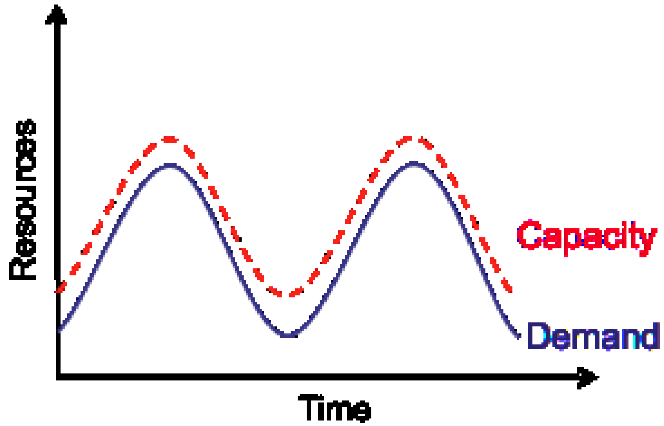
- Demand for a service varies with time
  - e.g., Peak loads
- Demand is unknown in advance
  - e.g., for new startup
- Batch analytics
  - e.g., 1000 EC2 instances for one hour cost the same as one instance for 1000 hours

Traditional Datacenter vs Cloud

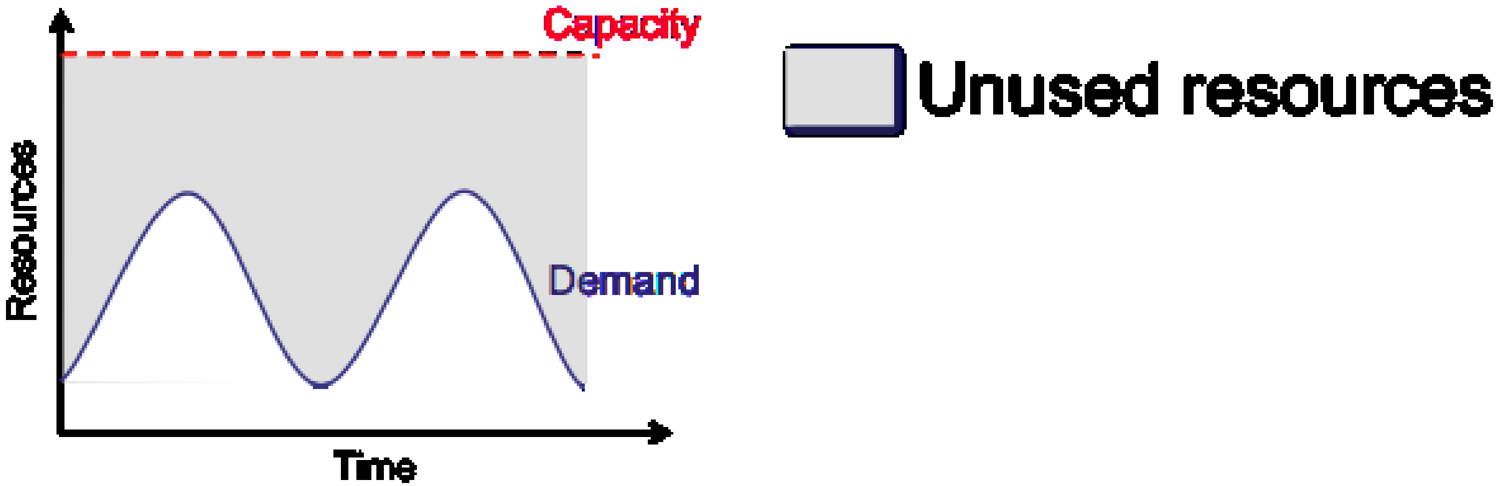
- Traditional datacenter



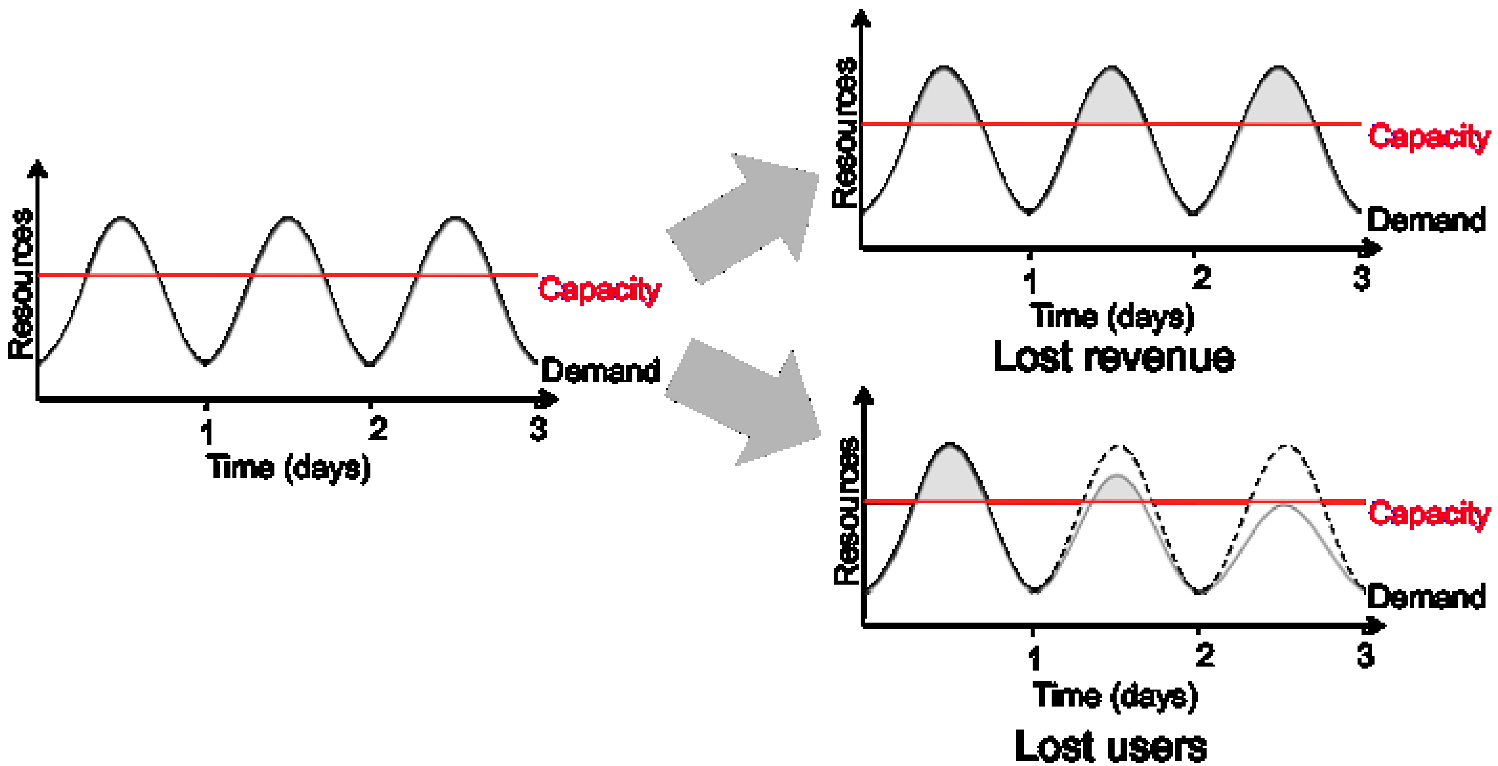
- Virtual datacenter in the cloud



Risk of Overprovisioning



Risks of Underprovisioning



Definition

- According to the National Institute of Standards and Technology (NIST):
  - On-demand self services: Quick, automated rental of capacity using Web interfaces
  - Broad network access
  - Resource pooling: Use of virtualization techniques
  - Rapid elasticity: Virtually unlimited capacity and scalability
  - Measured service: Pay-as-you-go

NIST: 3 Service Models (1)

- Cloud Infrastructure as a Service (IaaS)
  - Deliver computer infrastructure as a service (Virtual Machines, storage, …)
  - Example: Amazon EC2, Amazon S3
- Cloud Platform as a Service (PaaS)
  - Deliver computing platform and solution stack as a service (execution environment/framework)
  - Example: Google App Engine
- Cloud Software as a Service (SaaS)
  - Example: ERP software as a service, Salesforce.com

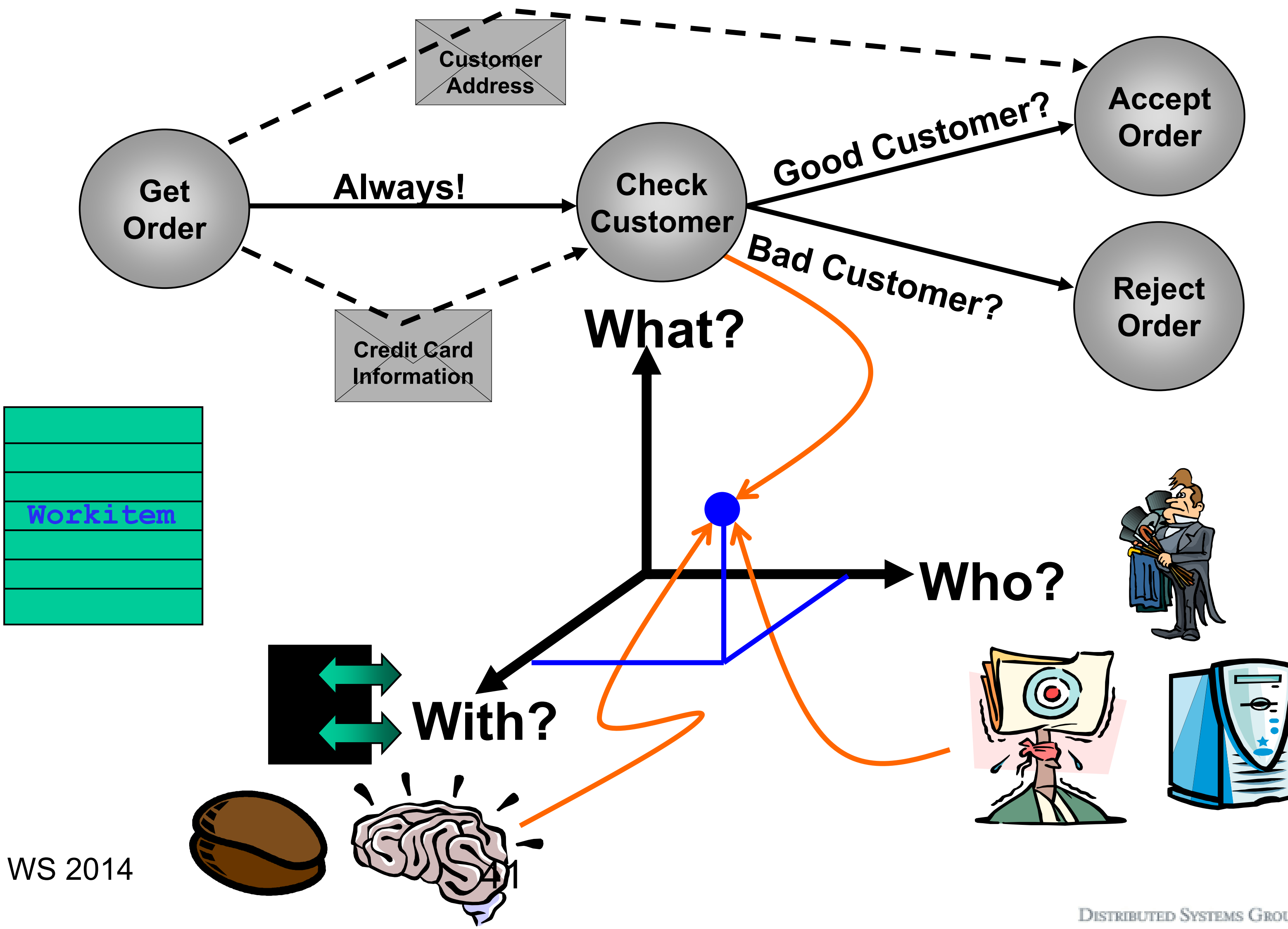
NIST: 4 Deployment Models

- Private Cloud: Operated solely for one single organization
- Community Cloud: Shared by several organizations
- Public Cloud: Open to general public, owned by an organization selling Cloud services
- Hybrid Cloud: Composition of two or more Cloud deployment models (private, community, public)

Outline

1. Overview
2. Peer-to-Peer Computing
3. Service-oriented Computing
4. Cloud Computing
5. Epilogue

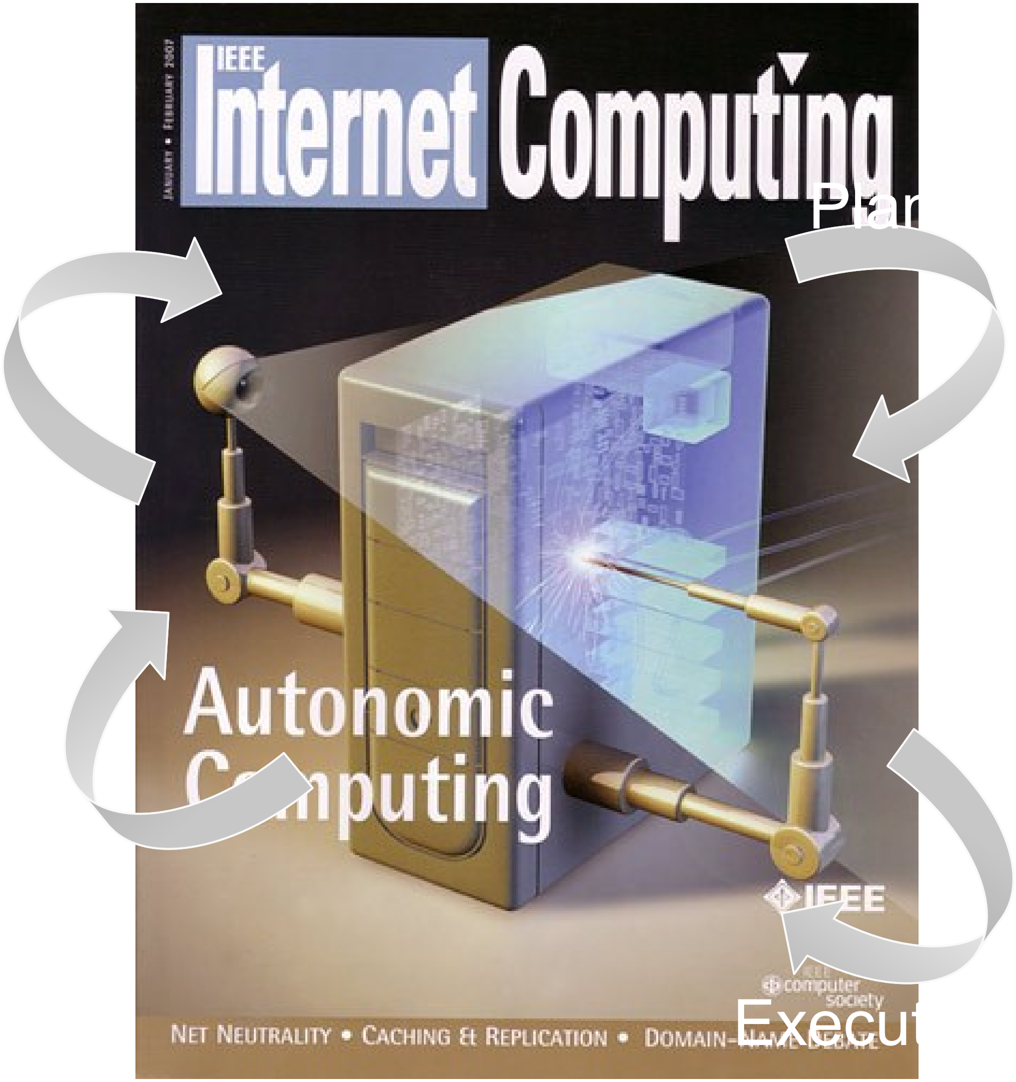
Business Processes and Workflows



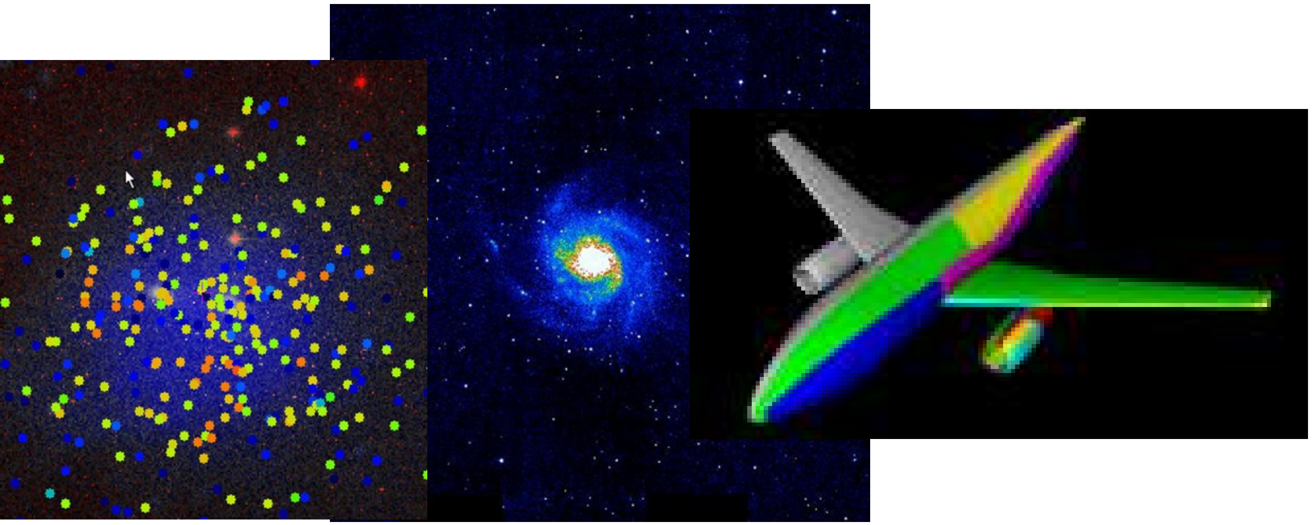
Mobile & Context-aware Computing

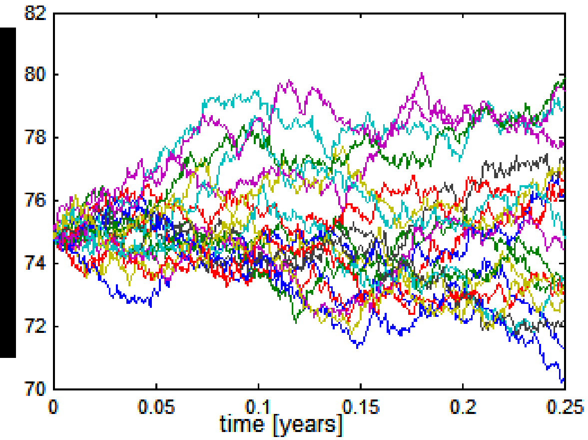
Autonomic Computing

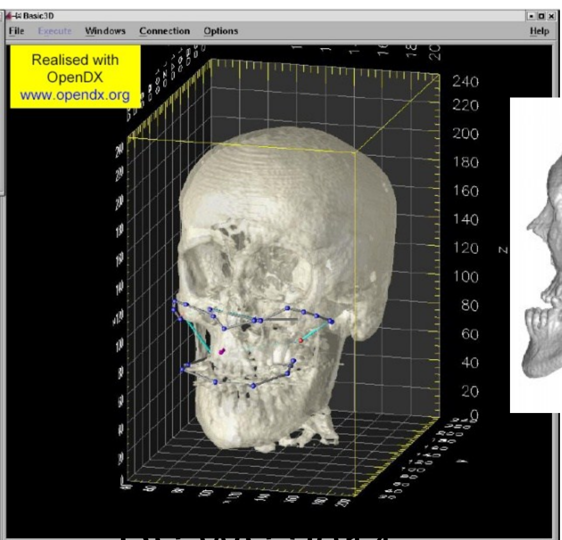
- Goals:
  - Self-Configuring
  - Self-Healing
  - Self-Optimizing
  - Self-Protecting

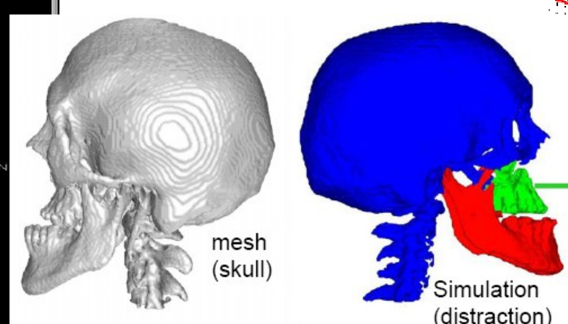


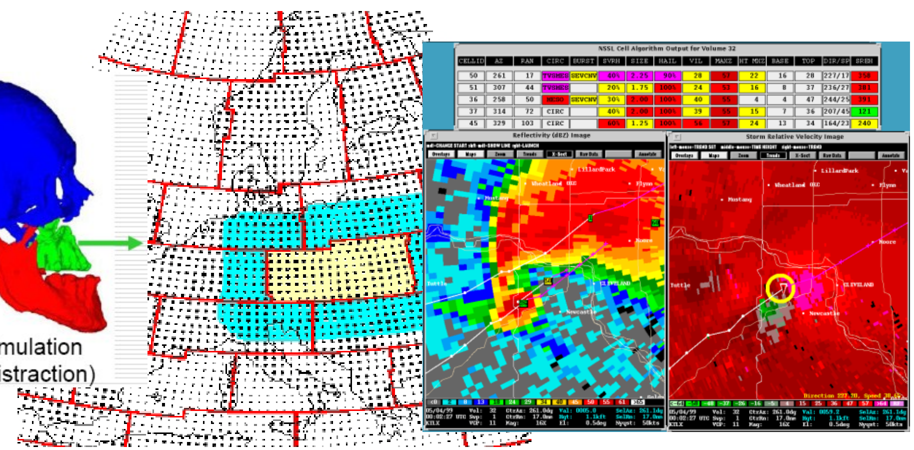
Scientific Computing











Cloud Computing

amazon web services™ SIMPLE MONTHLY CALCULATOR

Amazon Web Services » AWS Simple Monthly Calculator

Amazon S3 (US) ☑ Amazon S3 (EUR) ☐ Amazon EC2 ☐ Amazon SQS ☐

Amazon S3 (US)	Storage:	10	GB-months
	Data Transfer-in:	12	GB
	Data Transfer-out:	9	GB
	PUT/LIST Requests:	1000	Requests
	Other Requests:	10000	Requests
Amazon	Storage:	0	GB-months
	Data Transfer-in:	0	GB

Data Transfer-out: 0 GB

Calculate Now!

Customer Sample 1 (Amazon S3 only) · Customer Sample 2 (Amazon EC2 only) · Customer Sample 3 (Amazon SQS only) · Customer Sample 4 (Amazon S3 + EC2) · Customer Sample 5 (Amazon EC2 + SQS) · Customer Sample 6 (Amazon SQS + EC2 + S3)

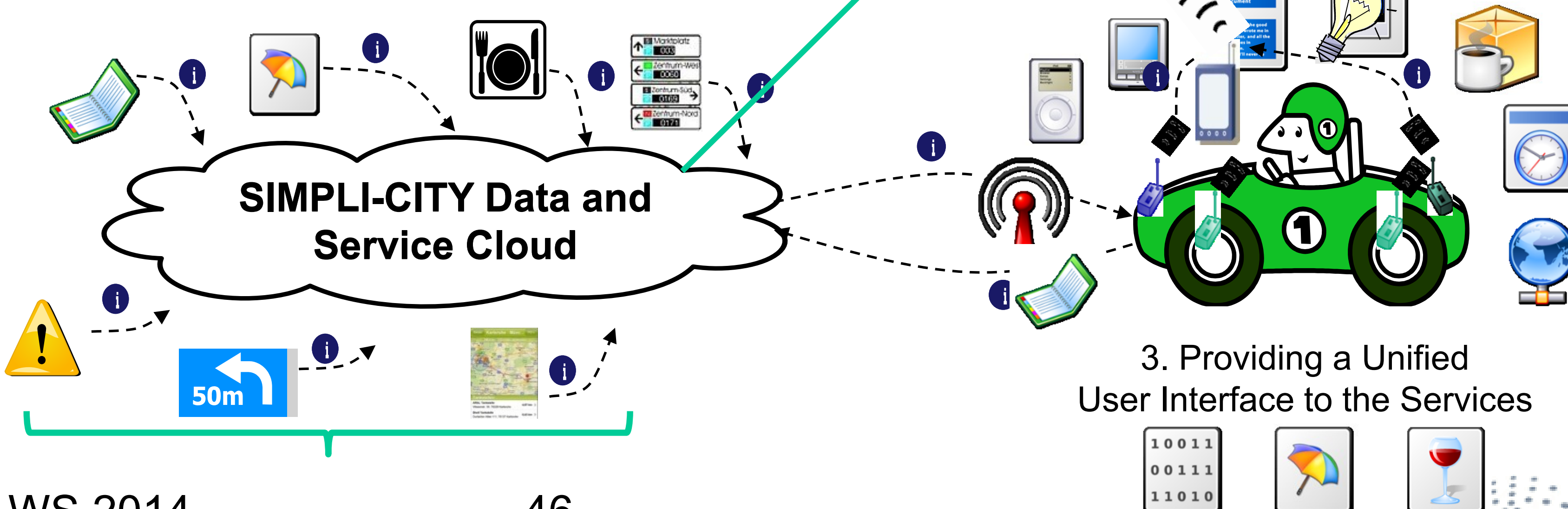
Estimate of Your Monthly Bill

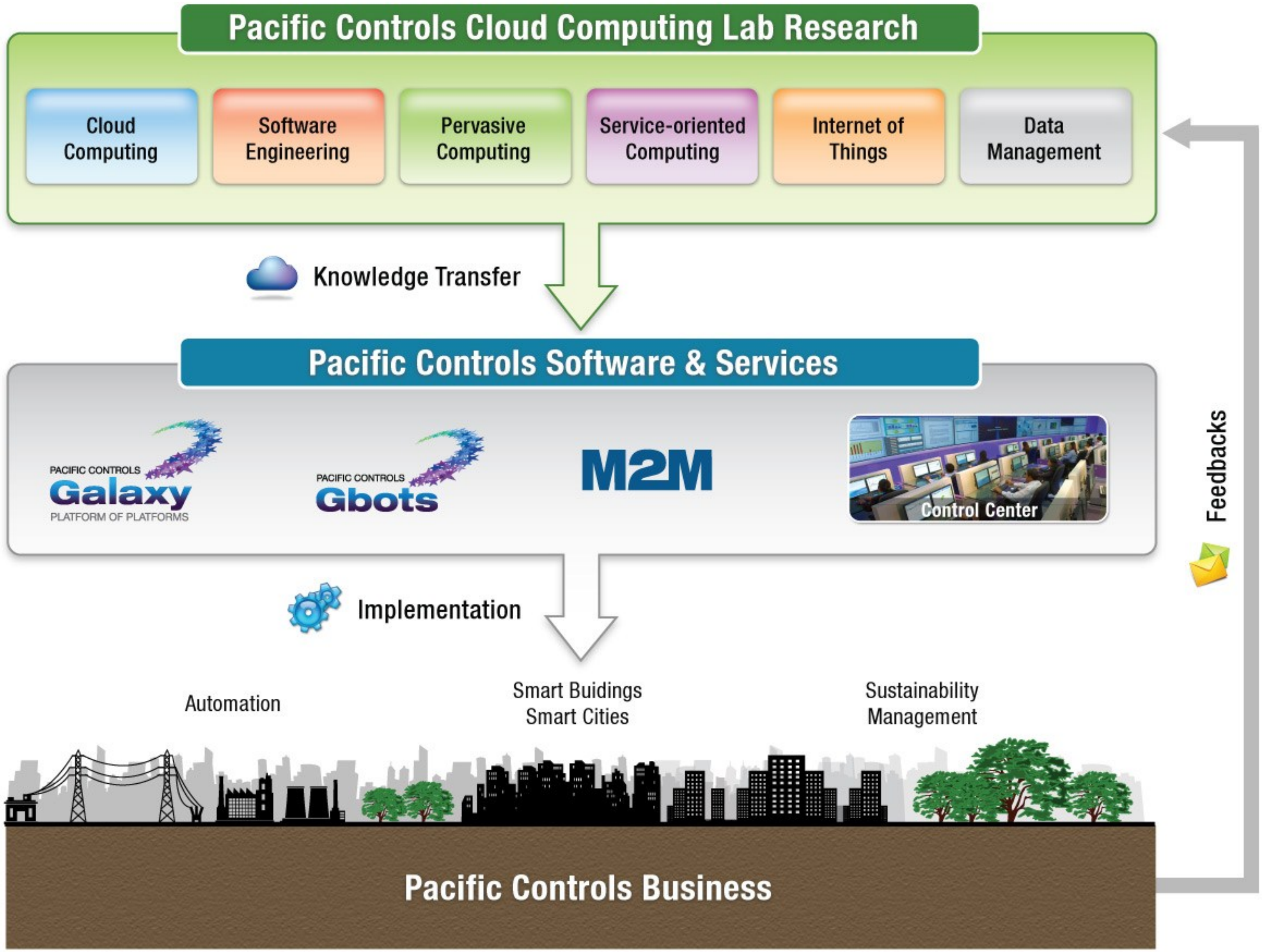
Amazon S3 (US)	Storage	$ 1.50	
	Data Transfer	$ 2.73	
	Requests	$ 0.02	
	Amazon S3 (US) Bill:		$ 4.25
Amazon S3 (EUR)	Storage	$ 0.00	
	Data Transfer	$ 0.00	
	Requests	$ 0.00	
	Amazon S3 (EUR) Bill:		$ 0.00
Amazon EC2	Compute	$ 0.00	
	Data Transfer	$ 0.00	
	EBS Volumes	$ 0.00	
	EBS Snapshots	$ 0.00	
	Amazon EC2 Bill:		$ 0.00
Amazon SQS	Messaging	$ 0.00	
	Data Transfer	$ 0.00	
	Amazon SQS Bill:		$ 0.00
Total Monthly Payment:			$ 4.25

Computing Power as a configurable, payable Service

The Road User Information System of the Future

- Ubiquitous Web access makes a multitude of information sources available to drivers:
  - Makes it difficult to get exactly the information I am looking for at the time I need it
- Vision: „SIRI for Mobility“





Pacific Controls Cloud Computing Lab Research
Cloud Computing
Software Engineering
Pervasive Computing
Service-oriented Computing
Internet of Things
Data Management
Knowledge Transfer
Pacific Controls Software & Services
PACIFIC CONTROLS Galaxy PLATFORM OF PLATFORMS
PACIFIC CONTROLS Gbots
M2M
Control Center
Implementation
Feedbacks
Automation
Smart Buildings
Smart Cities
Sustainability Management
Pacific Controls Business

Some Final Words

- We are always looking for motivated students:
  - Bachelor theses
  - Master theses
  - International internships
- Topics:
  - Internet of Things
  - Cloud Computing
  - Service-oriented Computing
  - Elastic Processes

Further Readings

- Armbrust et al.: A View of Cloud Computing, Communications of the ACM, 53(4), 2010.
- Papazoglou, Traverso, Dustdar, Leymann: Service-oriented Computing: State of the art and research challenges, Computer, 40(11), 2007.
- Steinmetz, Wehrle: Peer-to-Peer Systems and Applications, 2005.

Thanks for your attention!

Prof. Dr. Schahram Dustdar

Distributed Systems Group
TU Wien

dsg.tuwien.ac.at