DevOps and Architectural implications: Scalability

Fish
jfsc@cin.ufpe.br

THE ATTENDANTS WILL LEARN...

- The DevOps Pipeline
- What is the DevOps Pipeline?
- Stages;
- Moving through it.

- Software Architecture and DevOps;
- What is Software Architecture?
- Business goals+architecture+DevOps;
- Quality attributes;
- Tatics, Patterns and Styles;

- Service-oriented Development
- What is Microservices?
- Do we have other Approaches?
- How is DevOps related to Microservices?
- Conway's Law;

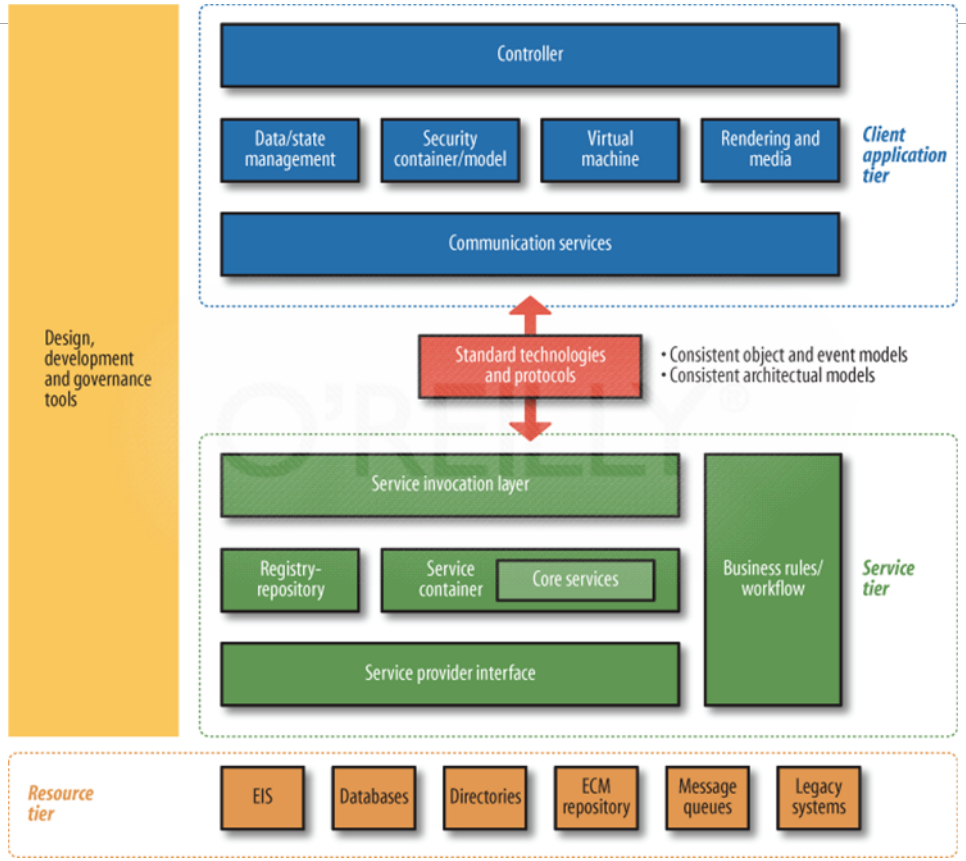
- What are the architectural structure implications of adopt Microservices in environments?
- Imutability
- Database;
- SOA vs Microservices;
- Tatics, Patterns, and Style;
- Quality Attributes;
- Scalability (the scale cube);
- PaaS;

What is software architecture?

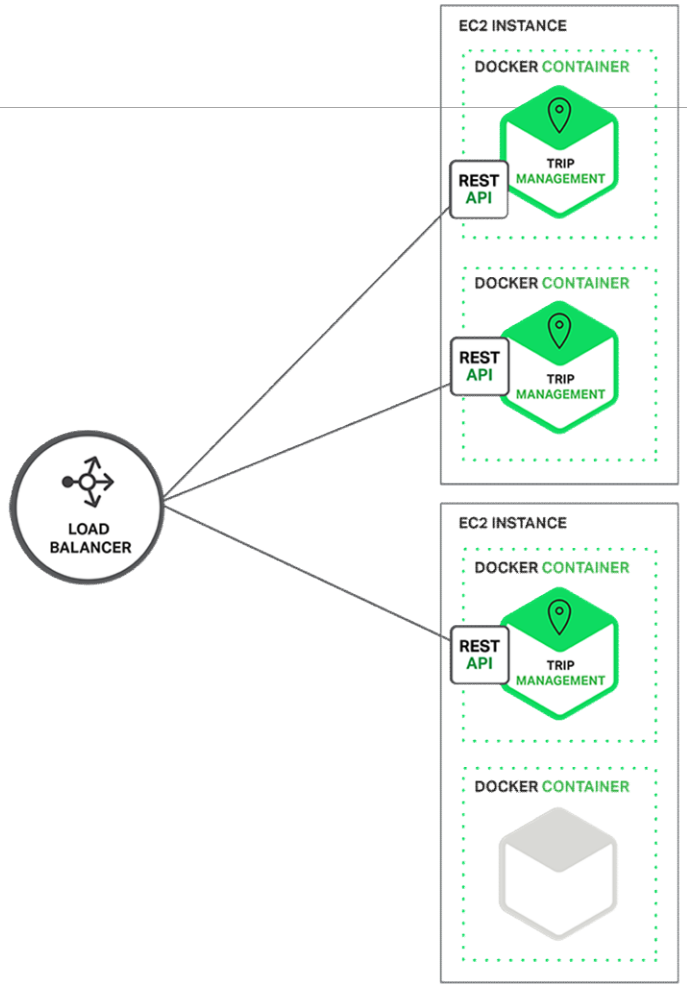
"is the set of structures needed to reason about the system, which comprises software elements, the relations among them, and the properties of both"

CMU Definition

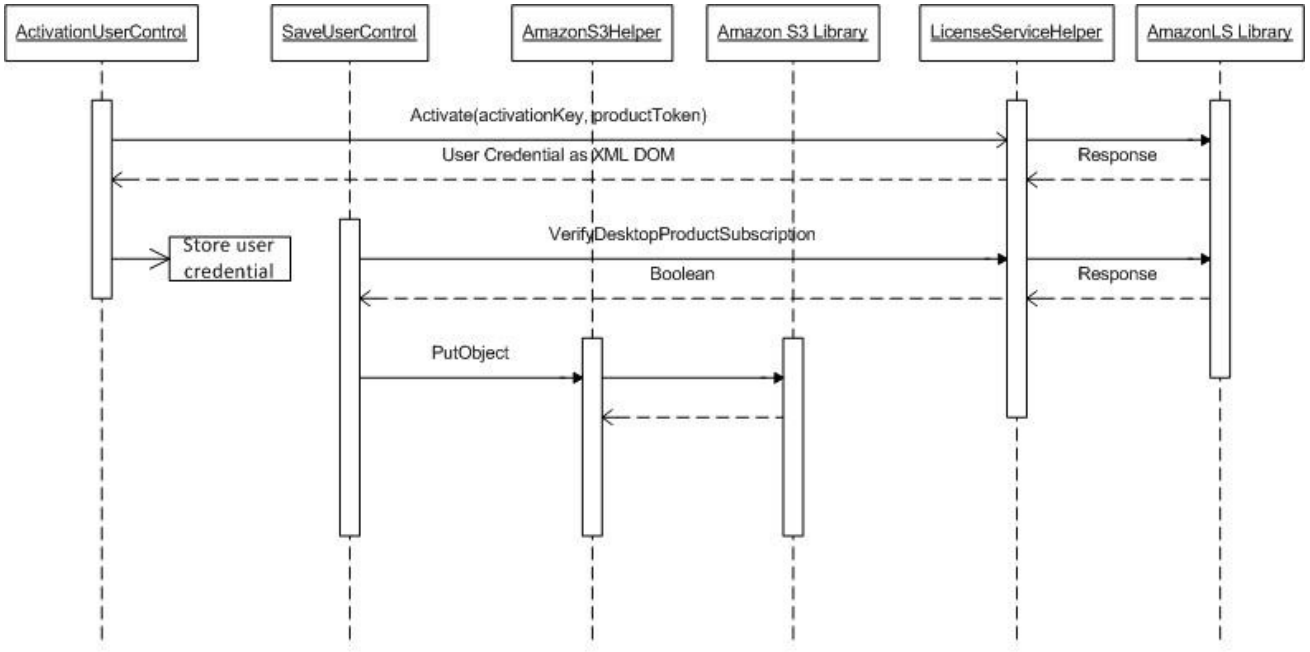
View: Static View



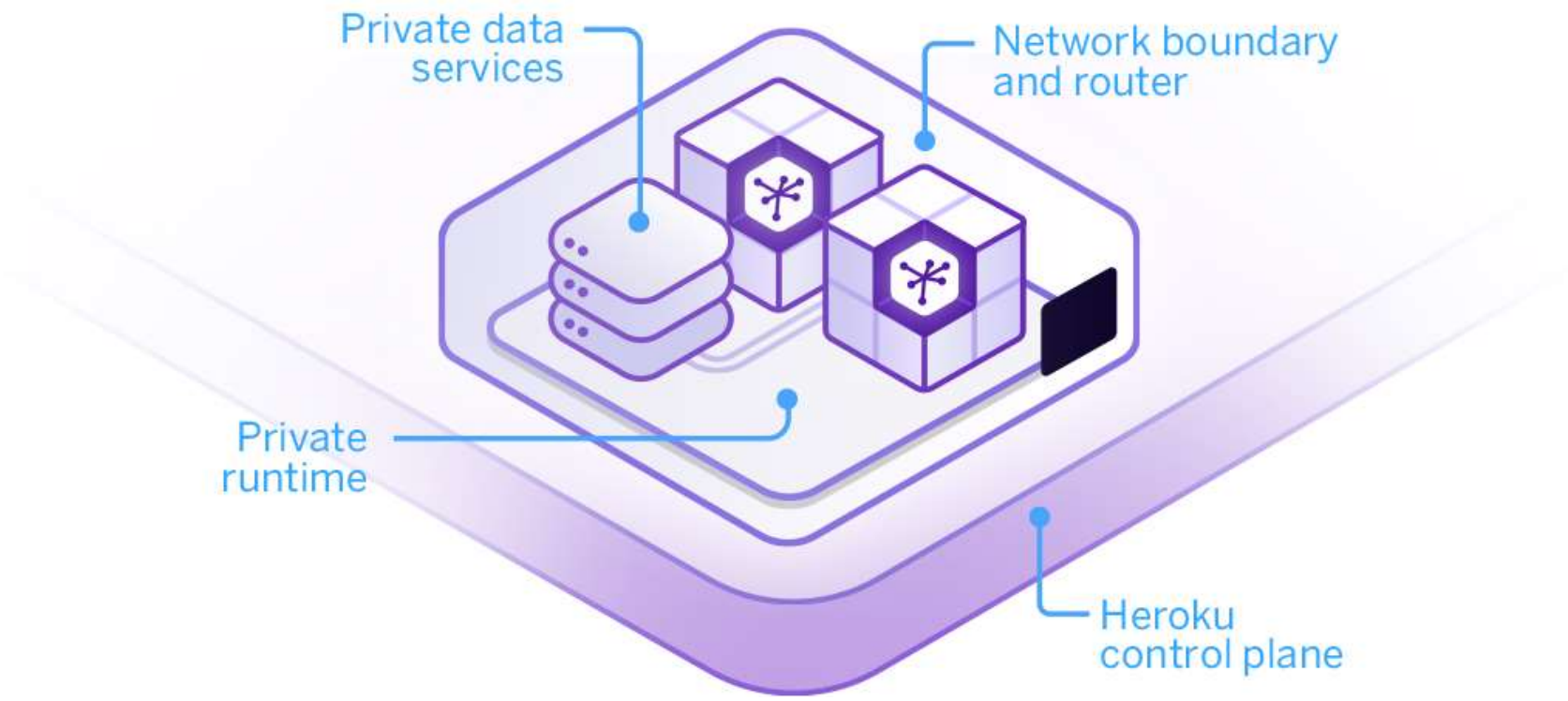
Views:
Deployment
View



View:
Dinamic



Heroku


Private data services
Network boundary and router
Private runtime
Heroku control plane

Business goals+architecture+DevOps

- WOW, this is software is very slow!
- Hey, my world of warcraft credentials were stolen!
- We will have 130.000 users in the system.

Business goals are related to quality atributes.

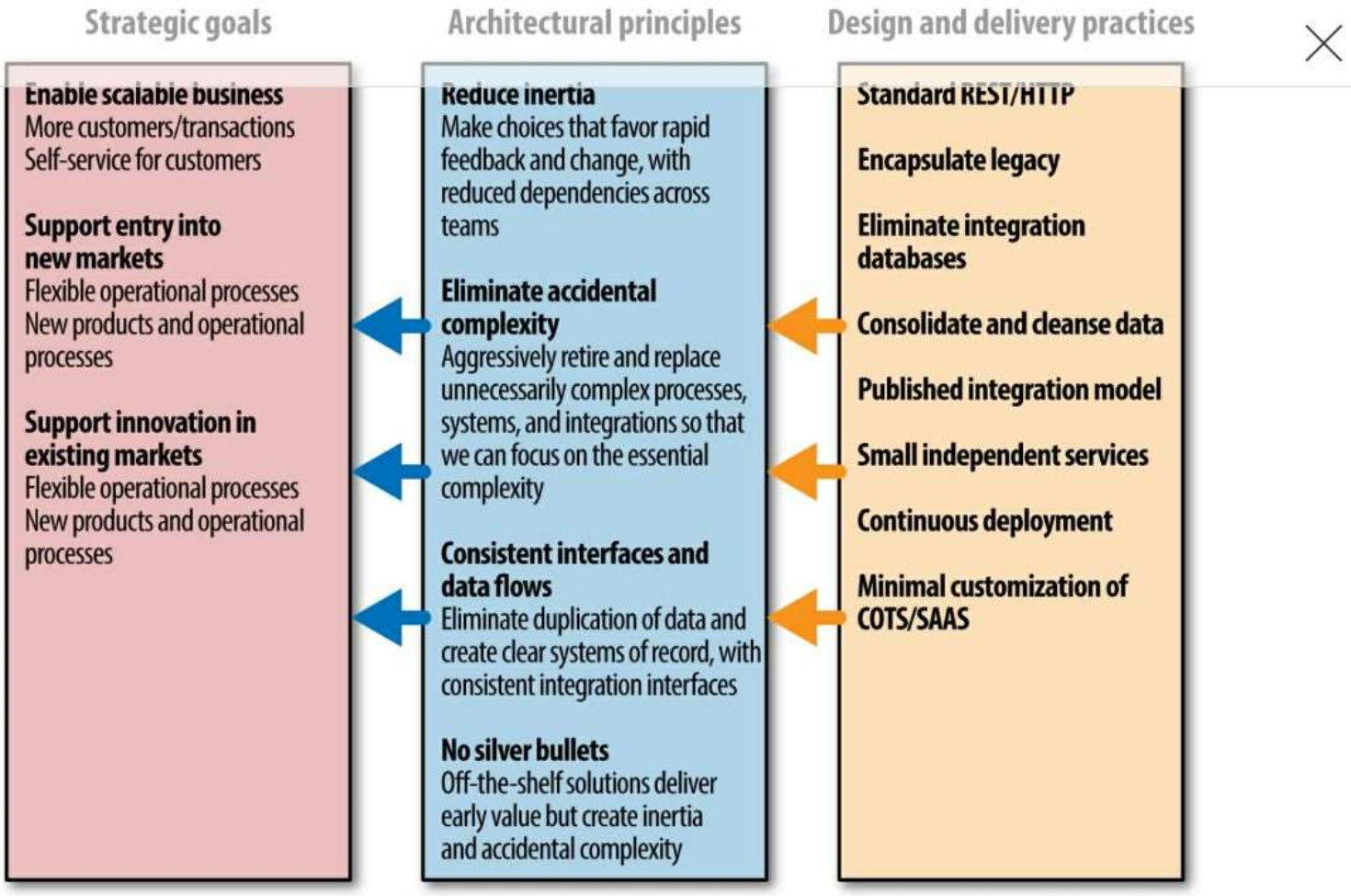
Quality

- Quality is the user perception of value;
- Depends on priority of Customer

Tradeoffs

- Safe vs performance;
- Portability vs maintainability

Trade-offs


Strategic goals
Architectural principles
Design and delivery practices
Enable scalable business
More customers/transactions
Self-service for customers
Support entry into new markets
Flexible operational processes
New products and operational processes
Support innovation in existing markets
Flexible operational processes
New products and operational processes
Reduce inertia
Make choices that favor rapid feedback and change, with reduced dependencies across teams
Eliminate accidental complexity
Aggressively retire and replace unnecessarily complex processes, systems, and integrations so that we can focus on the essential complexity
Consistent interfaces and data flows
Eliminate duplication of data and create clear systems of record, with consistent integration interfaces
No silver bullets
Off-the-shelf solutions deliver early value but create inertia and accidental complexity
Standard REST/HTTP
Encapsulate legacy
Eliminate integration databases
Consolidate and cleanse data
Published integration model
Small independent services
Continuous deployment
Minimal customization of COTS/SAAS

Scalability

Scalability is the ability to handle increased workload by repeatedly applying a cost effective strategy for extending a system's capacity.

CMU Definition

Peaks until unusable

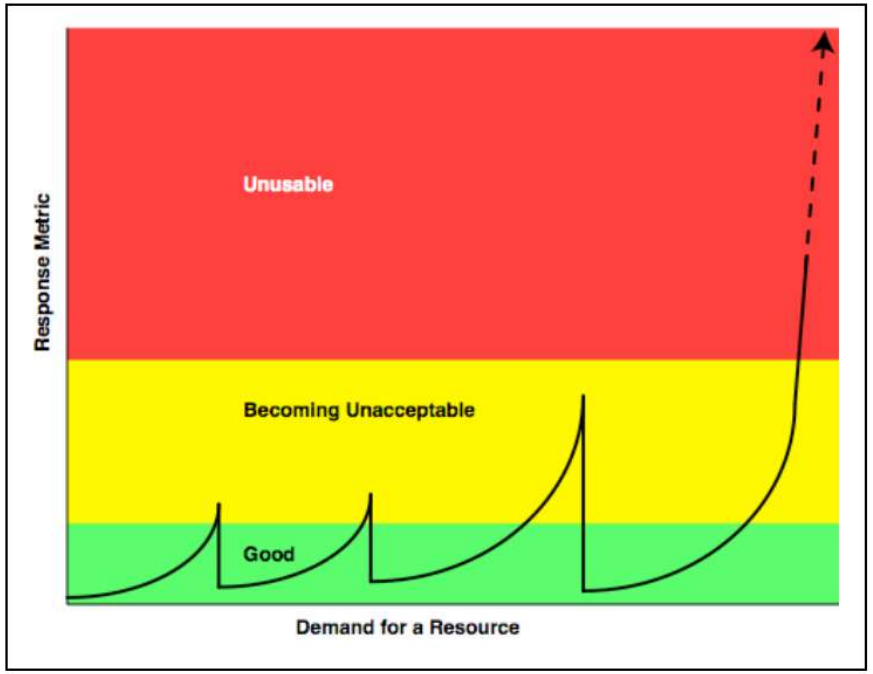
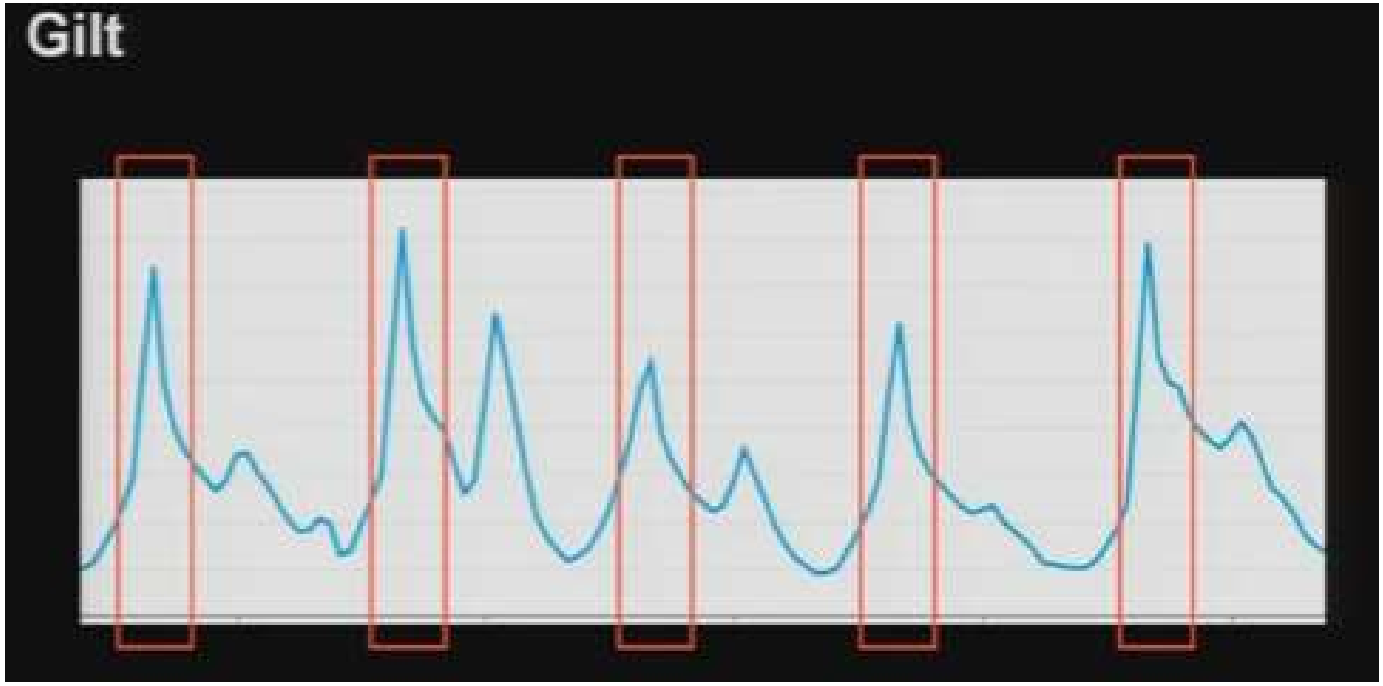


Figure 3: Response Metric vs. Demand for a Resource

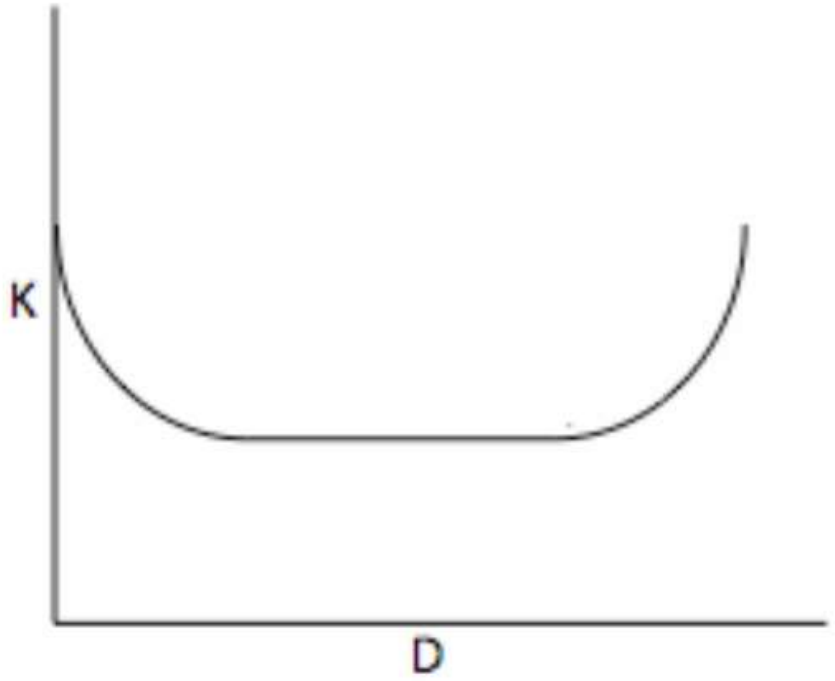
Spikes



Scalability Trade-off

- Performance
- Cost
- Operability
- Usability (functionality)
- Replica Consistency

In the end we want a “sweet spot”



Now, imagine all people inside a software system

	Facebook	Google+	Twitter	LinkedIn	Instagram	Pinterest	YouTube
Number of monthly active users (millions)	1,590[4]	300[5]	320[6]	400[7]	400[8]	100[9]	1,000+[10]

Phenomenas

- Unpredictable Expansions
- Frequent content generation
- Assymetric relationships (social network)

Impacts

- Storage (CAP);
- Network traffic;
- Energy Efficiency
- Hardware
- Software

Metrics

- Availability;
- Latency
- Interserve Communicatino
- Cost of Engineering and Resources
- Energy and Consuption and Maintanace Cost
- Internet Bandwidth Requiriments
- Data consistency
- Data Replication

Service-oriented Development

- Small services can be an answer;