End-to-End Service Performance and Dependability Analytics

Hong-Linh Truong
Faculty of Informatics, TU Wien

hong-linh.truong@tuwien.ac.at
http://www.infosys.tuwien.ac.at/staff/truong

Outline

- Fundamental
- End-to-end dependability in hybrid computing systems
- Summary

System function, behavior, structure and service

- Fundamental properties of a system
  - Functionality
  - Performance, dependability, security, cost
    - Called non-functional properties
  - Usability, manageability, adaptability/elasticity
- Structure of a system
  - A set of composite and atomic components
  - A composite component is composed of a set of components

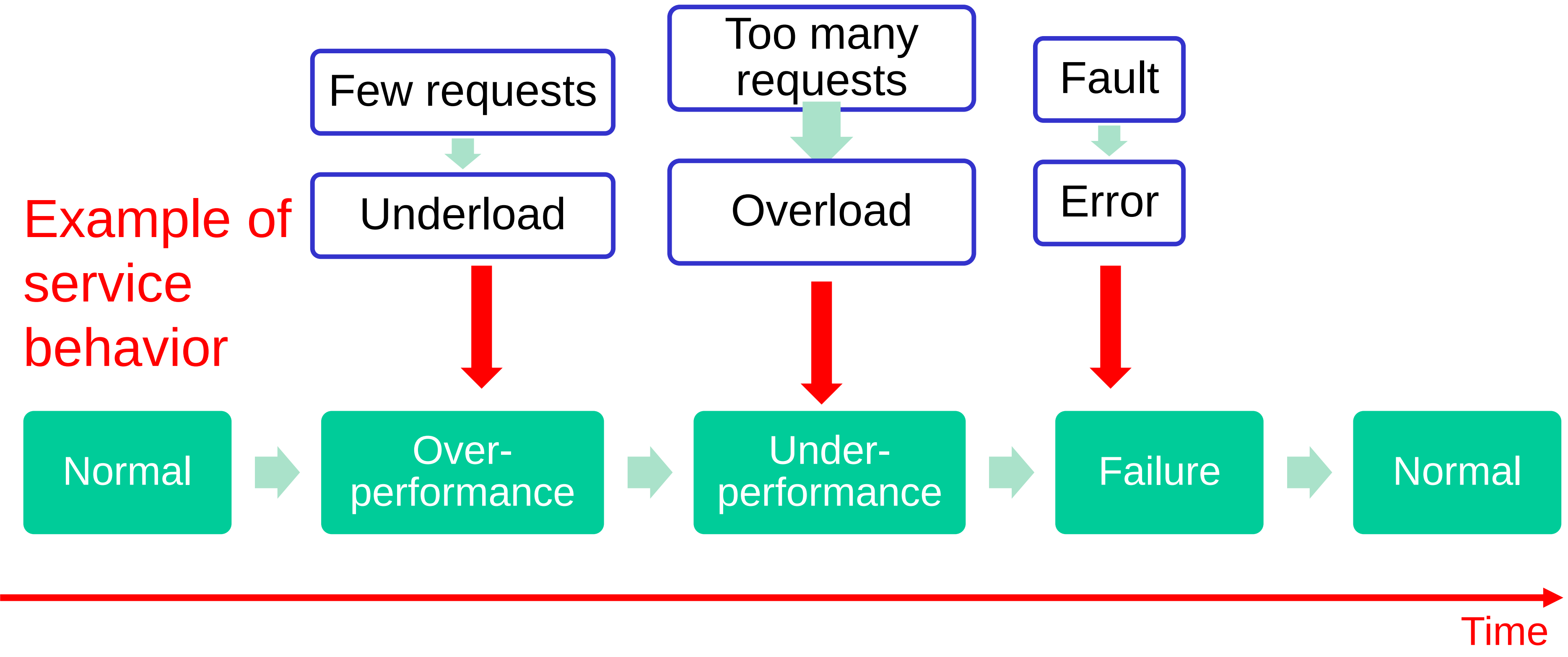
Client requirements/expectations

Clients require “correct service” w.r.t function and non-functional properties in an end-to-end view

Non-functional properties about performance, dependability, security and cost can be very subjective

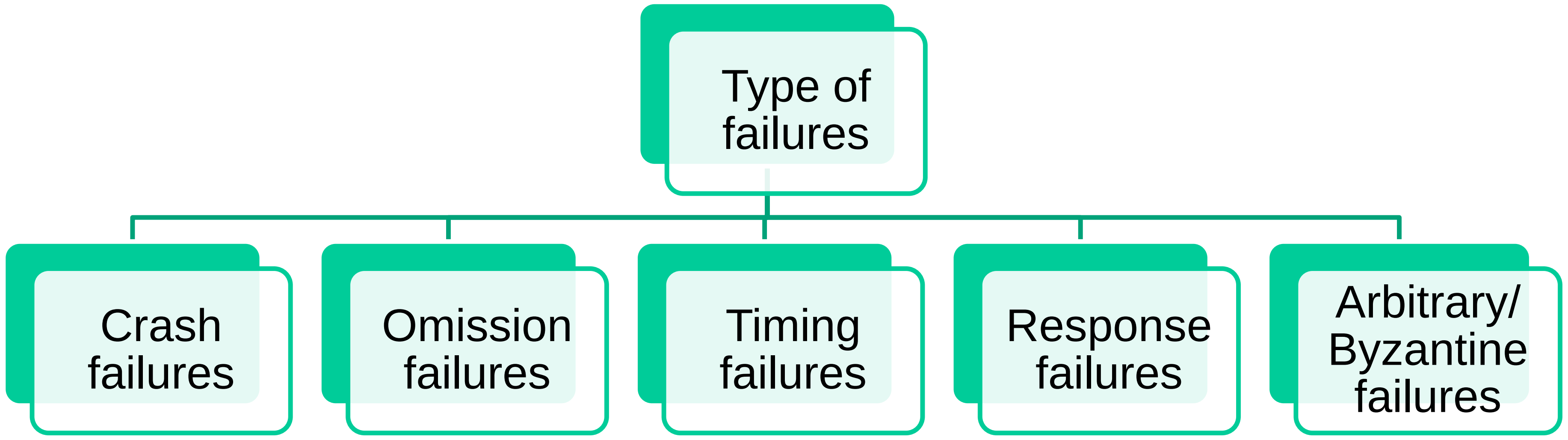
Check : John Knight, Fundamentals of Dependable Computing for Software Engineers, CRC Press, 2012

System behavior

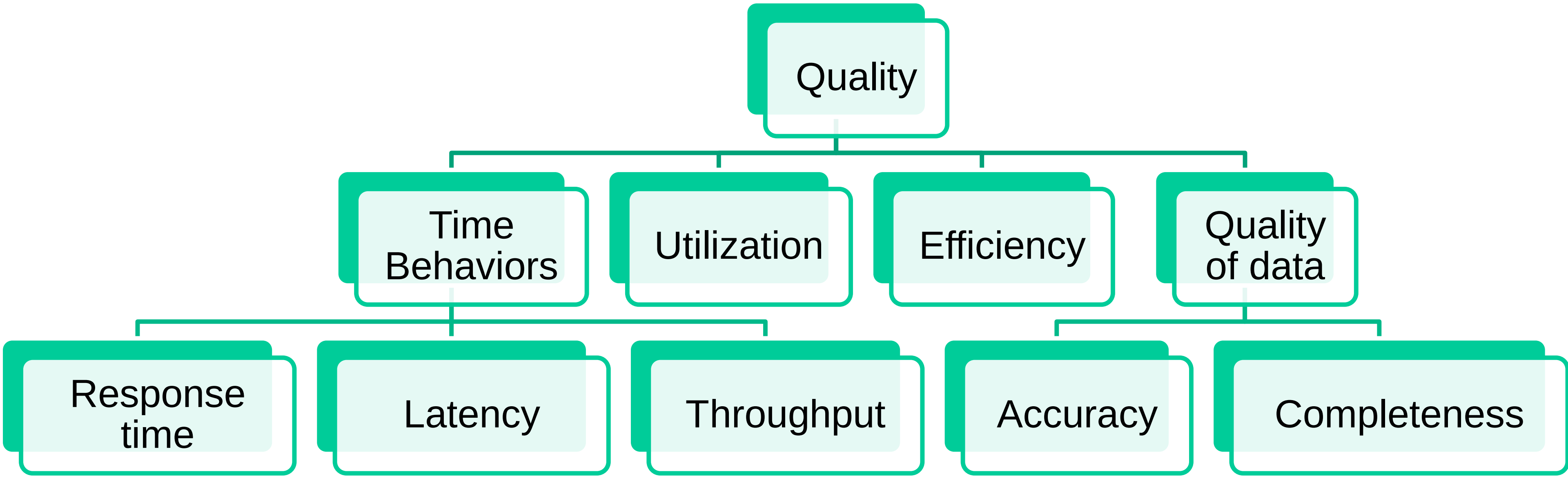


Normal: based on the service specification and design

Failure classification



Quality of service improvement



Industry view: https://guidingmetrics.com/content/cloud-services-industrys-10-most-critical-metrics/

NIST: https://www.nist.gov/sites/default/files/documents/itl/cloud/RATAX-CloudServiceMetricsDescription-DRAFT-20141111.pdf

Dealing with service failures and quality

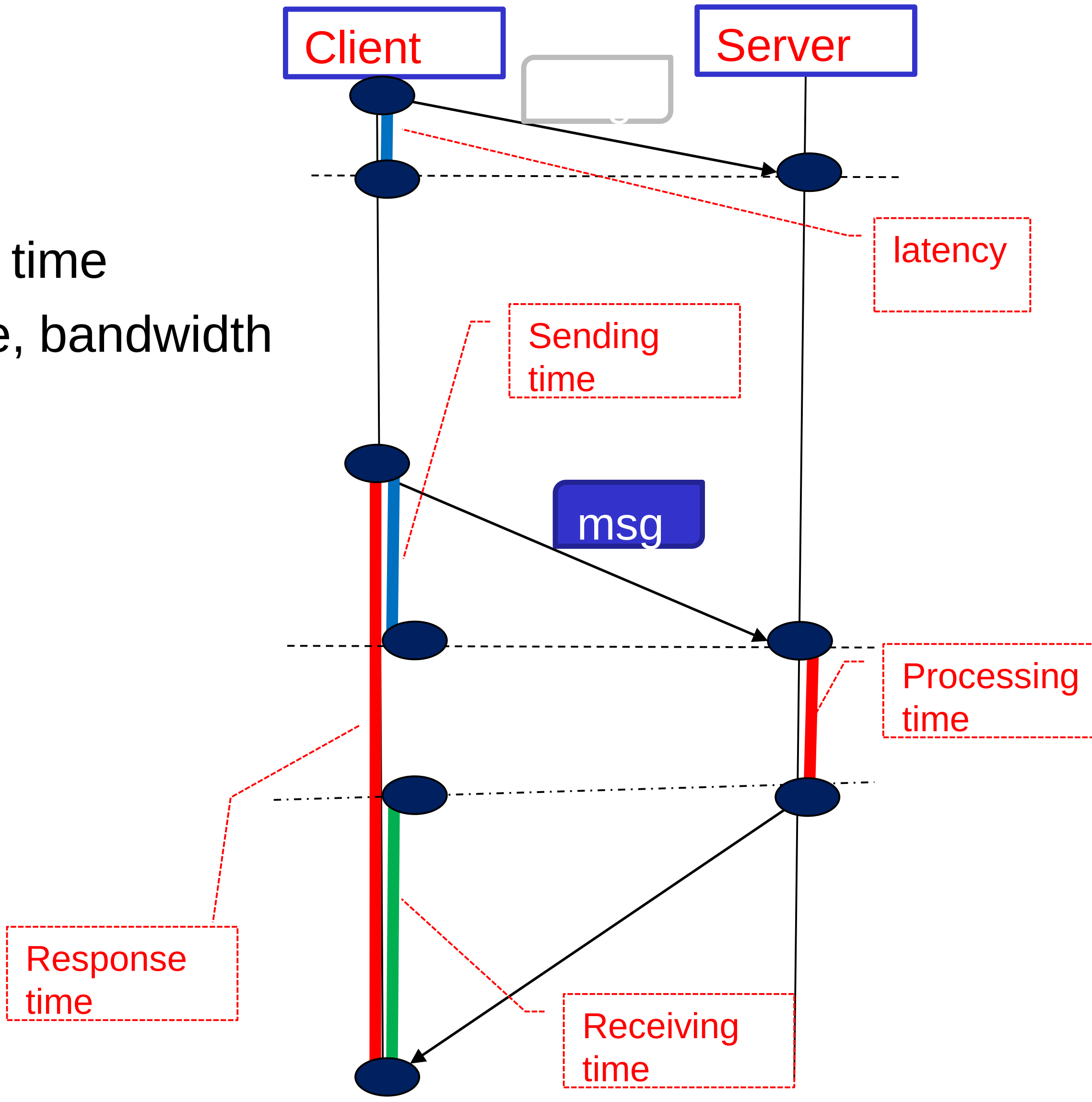
- Determines clearly system boundaries
  - The system under study, the system used to judge, and the environment
- Understands dependencies, e.g.
  - Among components in distributed systems
  - Single layer as well as cross-layered dependencies
- Determines types of metrics and failures and break down problems along the dependency path

Also check : John Knight, Fundamentals of Dependable Computing for Software Engineers, CRC Press, 2012

Performance metrics

Examples

- Timing behaviors
  - Communication
    - Latency/Transfer time
    - Data transfer rate, bandwidth
  - Processing
    - Response time
    - Throughput
- Utilization
  - Network utilization
  - CPU utilization
  - Service utilization
- Efficiency
- Data quality

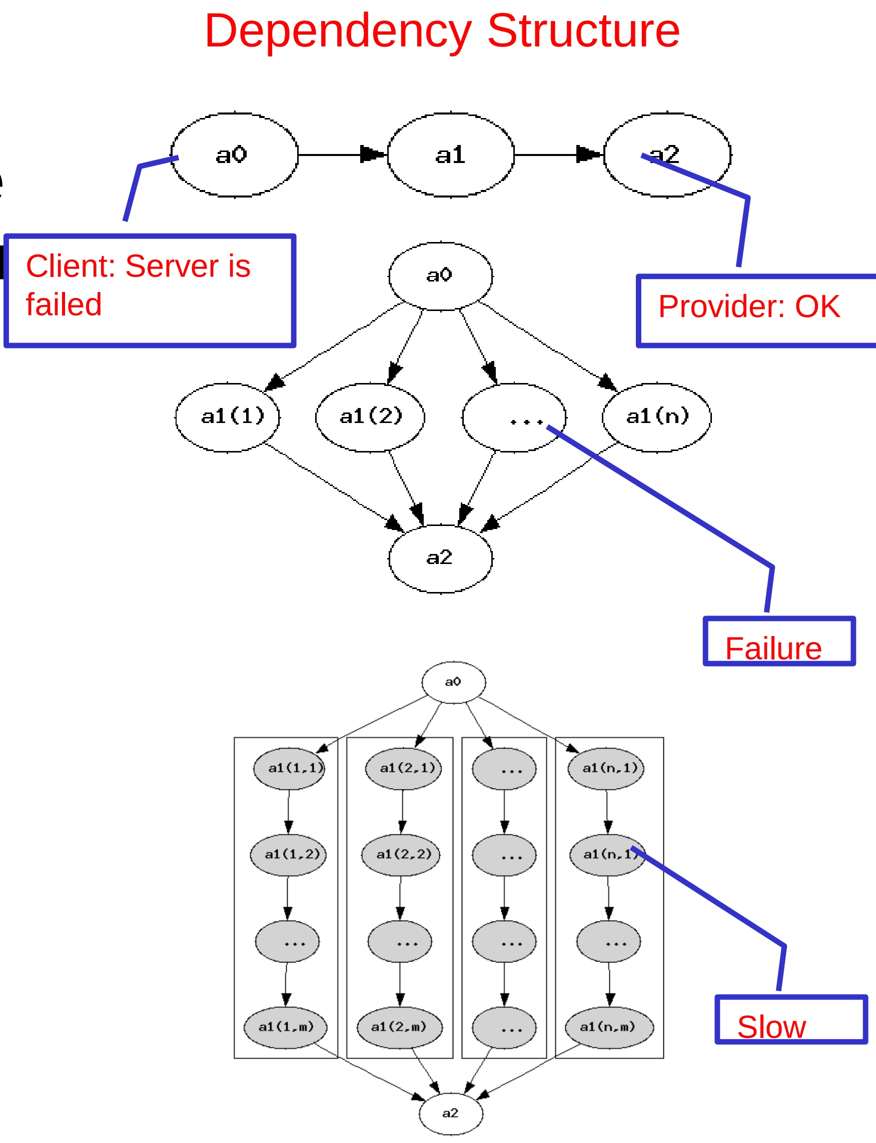


Measurement, Monitoring and Analysis

- Instrumentation and Sampling
  - Instrumentation: insert probes into systems so that you can measure system behaviors directly
  - Sampling: use components to take samples of system behaviors
- Monitoring
  - Probes or components perform sampling or measurements, storing and sharing measurments
- Analysis
  - Evaluate and interpret measurements for specific contexts
  - Can be subjective!

Composable methods and views

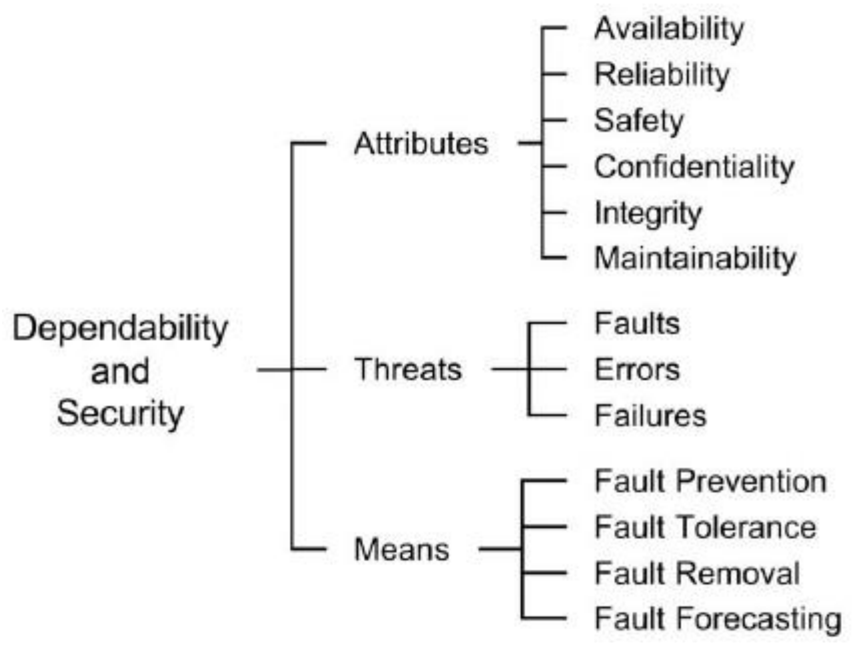
- Composable method
  - Divide a complex structure into basic common structu
  - Each basic structure has different ways to analyze specific failures/metrics
- Interpretation based on context/view
  - Client view or service provider view?
  - Conformity versus specific requirement assessment

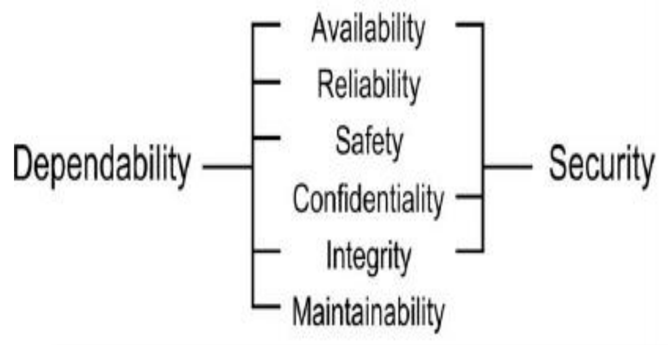


Performability

- What happens if the performance is unacceptable, e.g., the service cannot be scaled, the service is unreliable
- Technically, the system may still deliver its function
  - it may fail to deliver the expected non-functional properties as well as its function may fail eventually
- Performability measures a system performance and its dependability
  - Performance is currently not an attribute of dependability

Dependability Attributes, Threats and Means





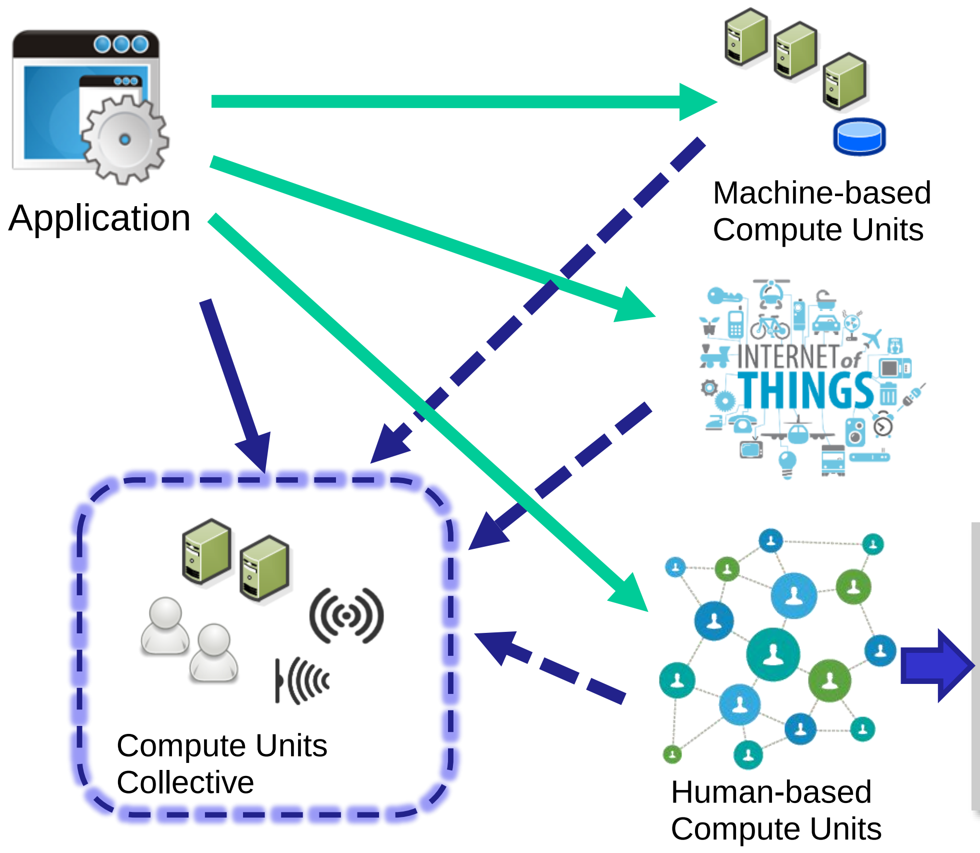
Sub mechanisms of resilience techniques

Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. 2004. Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Trans. Dependable Secur. Comput. 1, 1 (January 2004), 11-33.

Personal note: Performance should be an attribute as well!

System View

- Smart Cities
- IoT applications
- Predictive maintenance
- System optimization



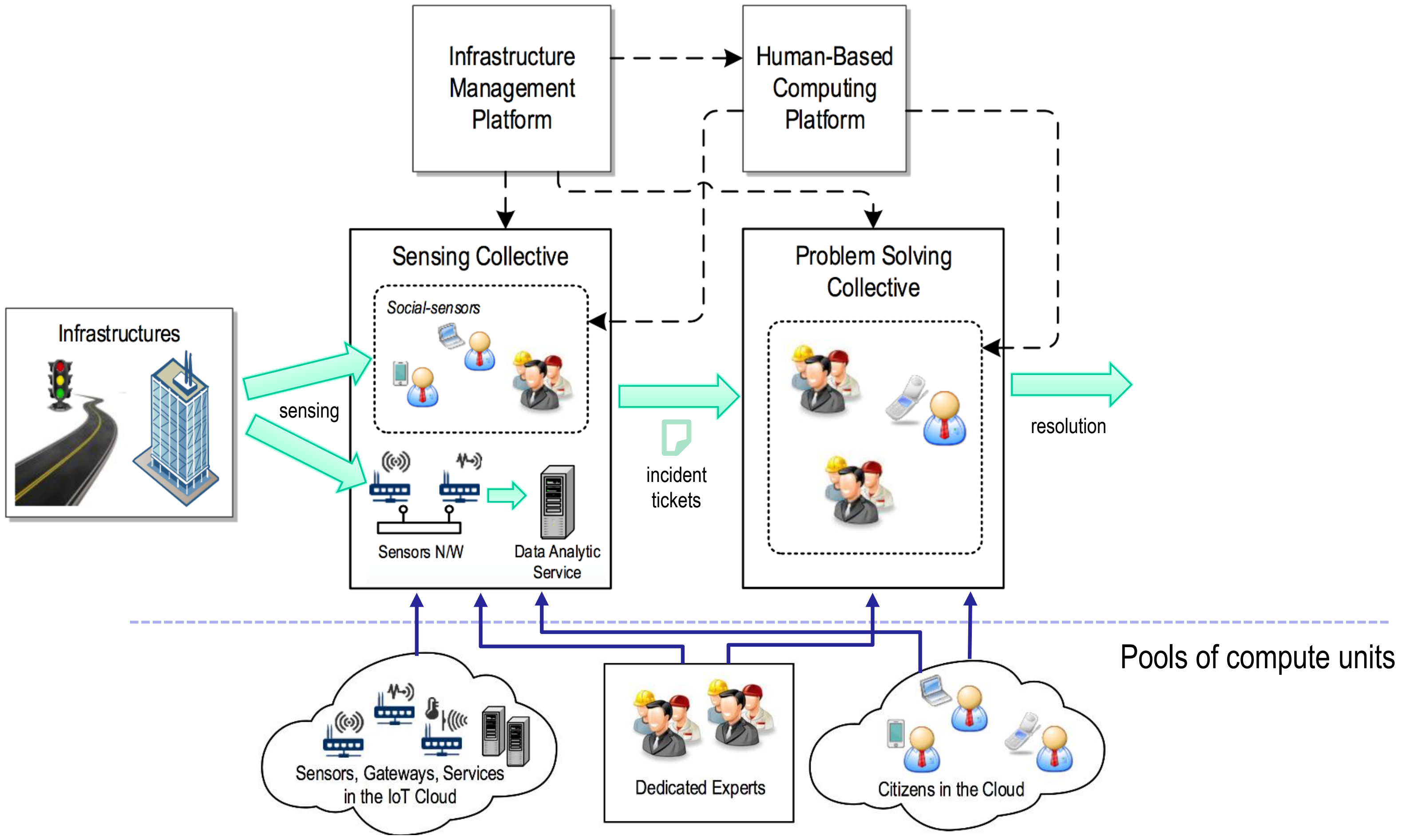
- Big data analytics
- Cloud services

- IoT infrastructures
- Sensing and actuating

- Crowdsourcing platforms
- Collective intelligence
- Social networks of experts
- On-premise domain experts

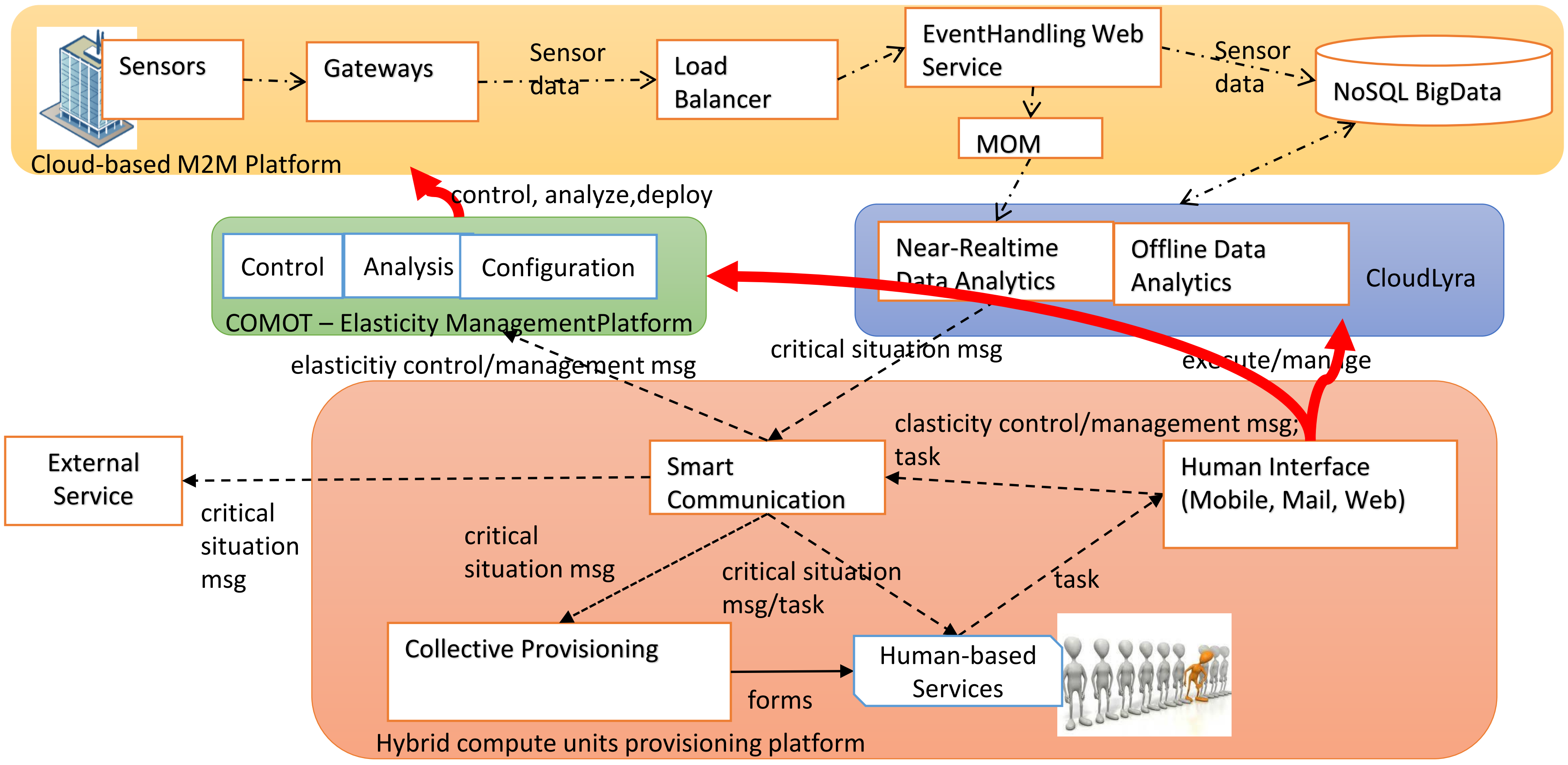
Link: http://dsg.tuwien.ac.at/staff/truong/publications/2016/truong-soca-panel-2016.pdf

Scenario/Application View

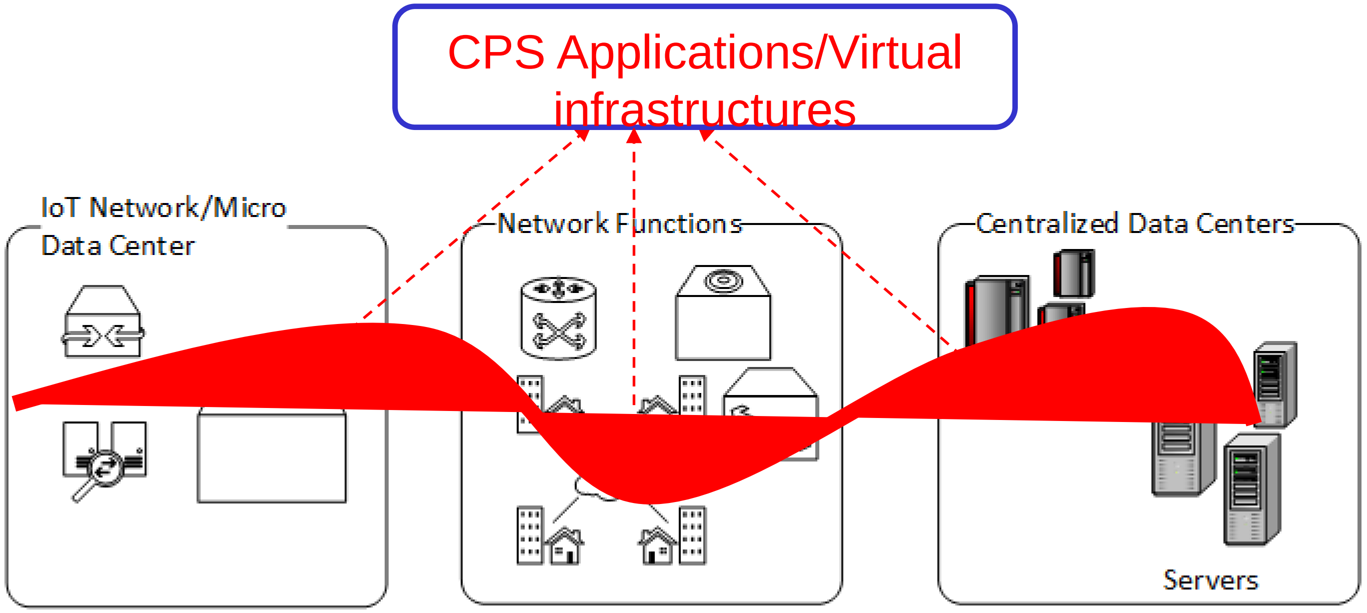


Link: http://www.infosys.tuwien.ac.at/staff/truong/publications/2016/truong-soca-panel-2016.pdf

Middleware View



View: end-to-end resource slice


CPS Applications/Virtual infrastructures
IoT Network/Micro Data Center
Network Functions
Centralized Data Centers
Servers

End-to end Resource slice

http://sincconcept.github.io/

End-to-end dependability

- What does it mean end-to-end? Examples?
  - Reflect the entire system
  - E.g., data reliability: from sensors to the final analytics results
- The user expects end-to-end dependability
  - E.g., specified in the expected QoR
- Providers/operators want to guarantee end-to-end dependability
  - Need to monitor different parts, each has subsystems/components
  - Coordination-aware dependability assurance
    - Autonomic and/or elasticity principles

Dependability in the cloud

- Infrastructure dependability and software dependability
- Fact: failures are inevitable! Why?
  - A lot of customers with different requirements
  - A lot data and services
  - Software are developed and deployed in a very short cycle.
- Design perspective: accept failures and think how to deal with failures

Kashi Venkatesh Vishwanath and Nachiappan Nagappan. Characterizing cloud computing hardware reliability. In Proceedings of the 1st ACM symposium on Cloud computing (SoCC '10). ACM,193-204.

Dai, Y. S., Yang, B., Dongarra, J., Zhang, G.: Cloud Service Reliability: Modeling and Analysis. In: The 15th IEEE Pacific Rim International Symposium on Dependable Computing (2009) - http://www.netlib.org/utk/people/JackDongarra/PAPERS/Cloud-Shaun-Jack.pdf

Hiranya Jayathilaka, Chandra Krintz, Rich Wolski:
Performance Monitoring and Root Cause Analysis for Cloud-hosted Web Applications. WWW 2017: 469-478

Dealing with dependability problems

- Traditional techniques
  - E.g., Replication & Redundancy
- Virtualization
  - Hide dependability problems and allow quick recovery through virtualization techniques
- Elasticity
  - Compensate dependability problems with elasticity of resources, costs and quality

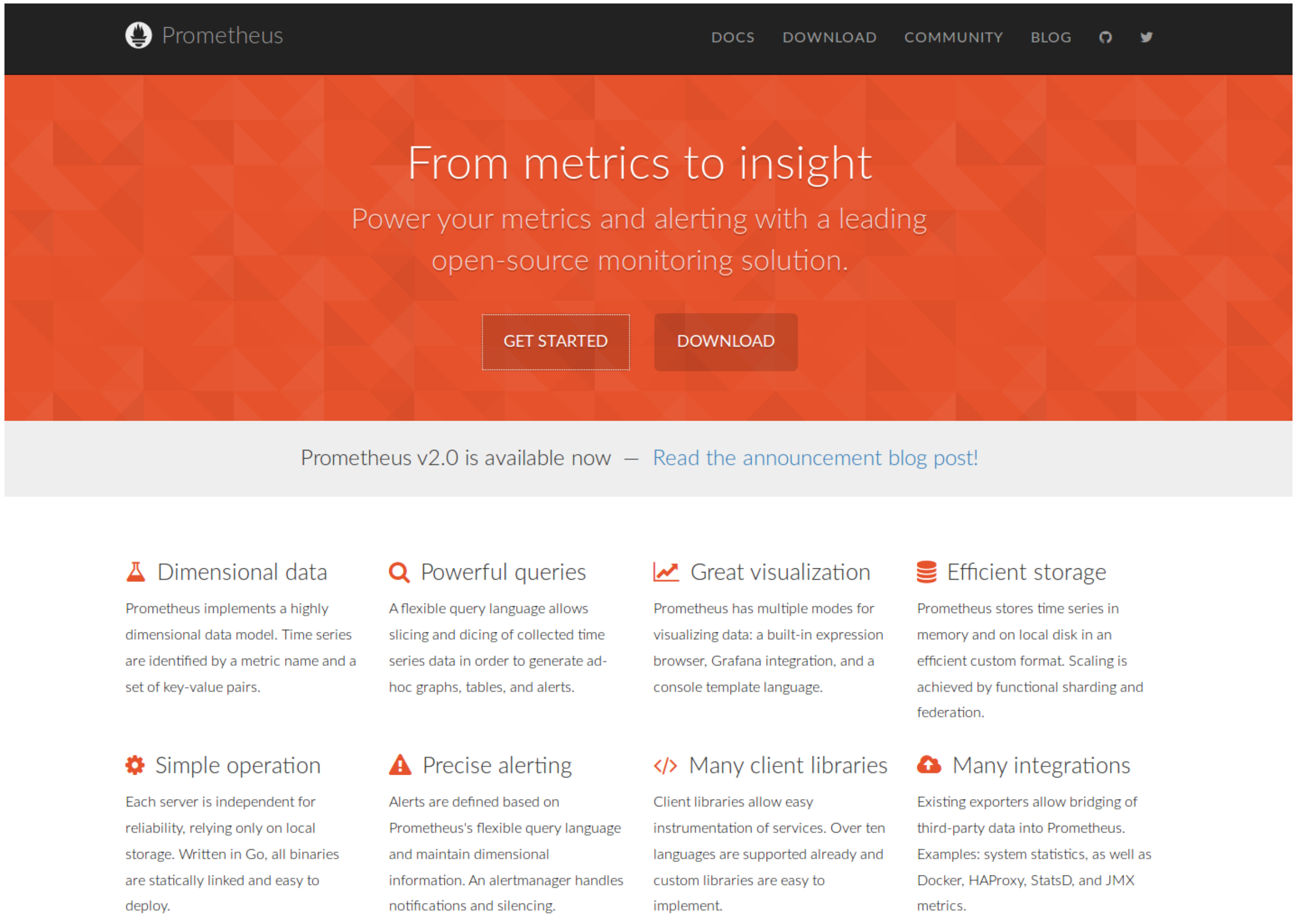
“Site Reliability Engineering: How Google Runs Production Systems”: https://landing.google.com/sre/

Dependability for IoT

- Different levels
  - Infrastructures: Things and communication networks
  - Software: sensors, gateways, and actuators,
- Computation dependability versus protocols versus data dependability
- Well-established work
  - Network dependability
- Not understood well
  - Data dependability and its impacts

Ivanovitch Silva, Rafael Leandro, Daniel Macedo, and Luiz Affonso Guedes. A dependability evaluation tool for the Internet of Things. Comput. Electr. Eng. 39, 7 (2013) - https://sigaa.ufrn.br/sigaa/verProducao?idProducao=1574768&key=7bb98e0e59f0d978abd5c578ae291e02

Monitoring Tools (1)



From metrics to insight

Power your metrics and alerting with a leading open-source monitoring solution.

Prometheus v2.0 is available now — Read the announcement blog post!

Dimensional data

Prometheus implements a highly dimensional data model. Time series are identified by a metric name and a set of key-value pairs.

Powerful queries

A flexible query language allows slicing and dicing of collected time series data in order to generate ad-hoc graphs, tables, and alerts.

Great visualization

Prometheus has multiple modes for visualizing data: a built-in expression browser, Grafana integration, and a console template language.

Efficient storage

Prometheus stores time series in memory and on local disk in an efficient custom format. Scaling is achieved by functional sharding and federation.

Simple operation

Each server is independent for reliability, relying only on local storage. Written in Go, all binaries are statically linked and easy to deploy.

Precise alerting

Alerts are defined based on Prometheus's flexible query language and maintain dimensional information. An alertmanager handles notifications and silencing.

Many client libraries

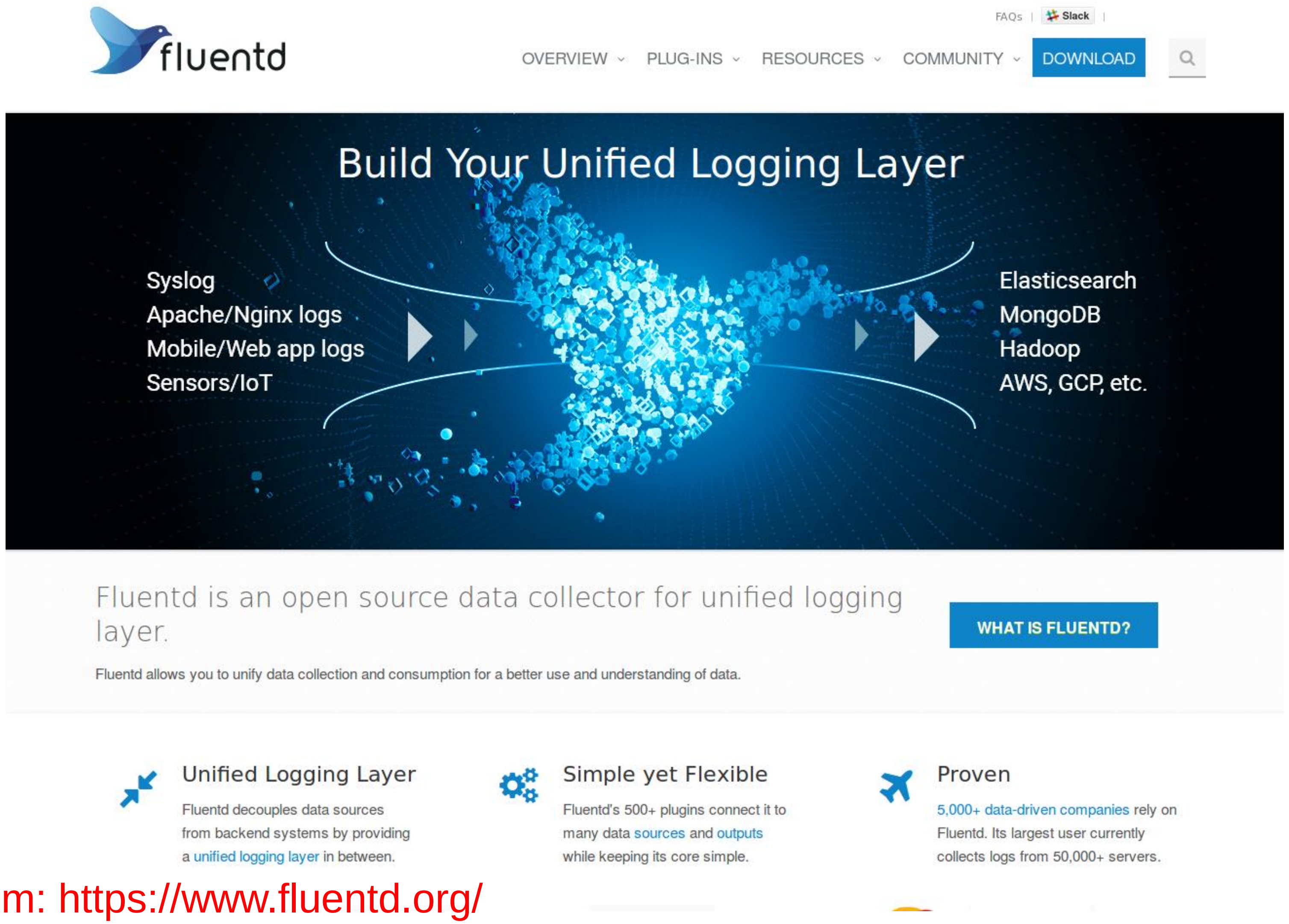
Client libraries allow easy instrumentation of services. Over ten languages are supported already and custom libraries are easy to implement.

Many integrations

Existing exporters allow bridging of third-party data into Prometheus. Examples: system statistics, as well as Docker, HAProxy, StatsD, and JMX metrics.

From: https://prometheus.io/

Monitoring Tools (2)



FAQs | Slack |

fluentd

OVERVIEW PLUG-INS RESOURCES COMMUNITY DOWNLOAD

Fluentd is an open source data collector for unified logging layer.

WHAT IS FLUENTD?

Fluentd allows you to unify data collection and consumption for a better use and understanding of data.

Unified Logging Layer

Fluentd decouples data sources from backend systems by providing a unified logging layer in between.

Simple yet Flexible

Fluentd's 500+ plugins connect it to many data sources and outputs while keeping its core simple.

Proven

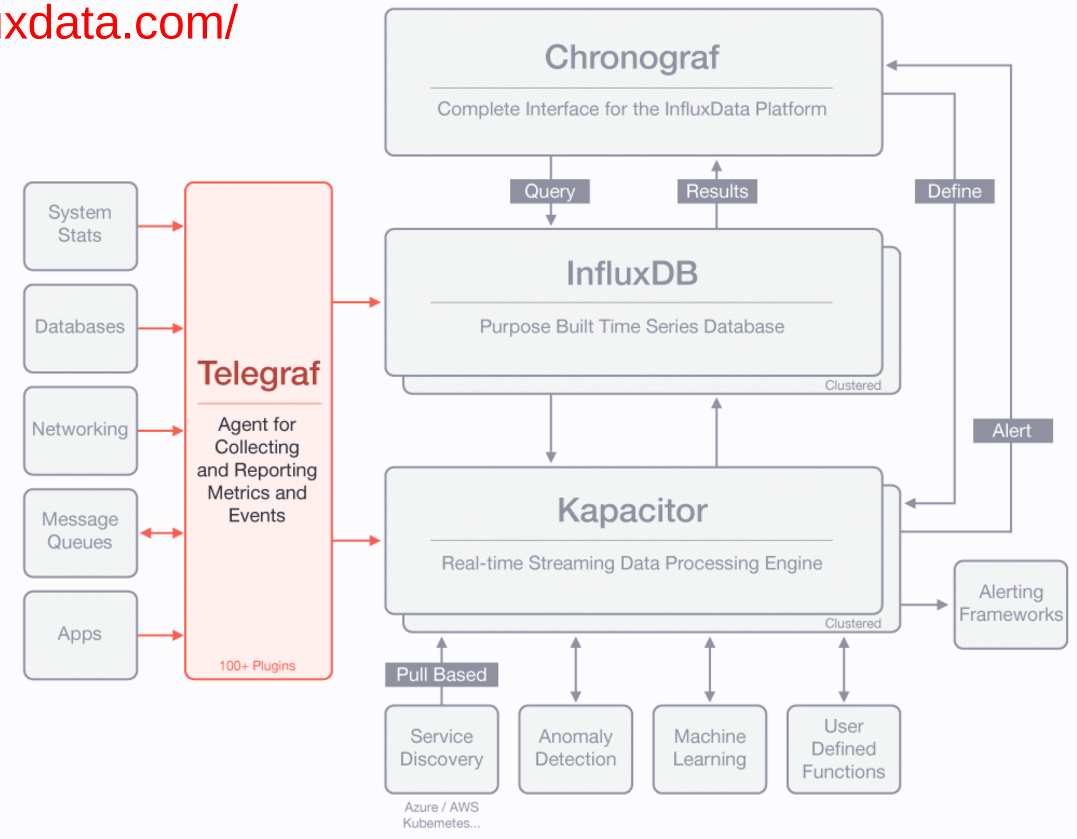
5,000+ data-driven companies rely on Fluentd. Its largest user currently collects logs from 50,000+ servers.

From: https://www.fluentd.org/

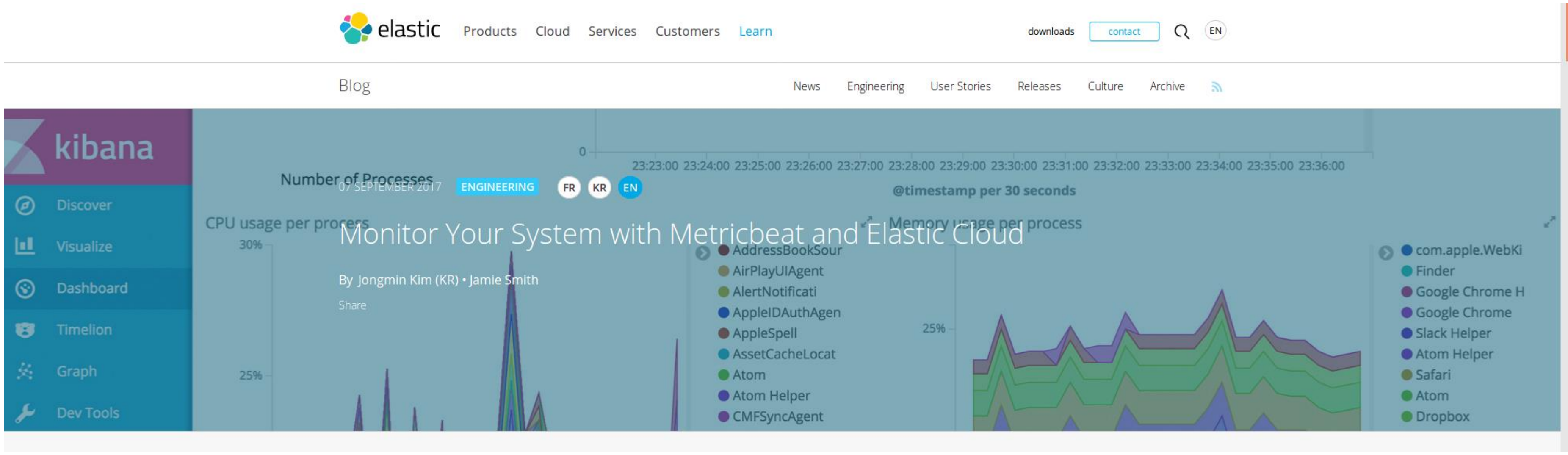
Monitoring Tools (3)

It is part of the TICK stack and is a plugin-driven server agent for collecting and reporting metrics. Telegraf has plugins or integrations to source a variety of metrics directly from the system it's running on, pull metrics from third-party APIs, or even listen for metrics via a StatsD and Kafka consumer services. It also has output plugins to send metrics to a variety of other datastores, services, and message queues, including InfluxDB, Graphite, OpenTSDB, Datadog, Librato, Kafka, MQTT, NSQ, and many others.

From: https://www.influxdata.com/



Monitoring Tools (4)



07 SEPTEMBER 2017 ENGINEERING FR KR EN

Monitor Your System with Metricbeat and Elastic Cloud

By Jongmin Kim (KR) • Jamie Smith

Share

From: https://www.elastic.co/

Your next assignment

1. Tools and end-to-end metrics for IoT Cloud Systems
   - Which are interesting metrics (and tools for analysis)
   - Technical debt and performance/dependability
   - Instrumentation and (micro)service/system engineering
2. Too much monitoring data
   - How can machine learning help for performance and dependability of IoT Cloud systems?
   - How can big data analytics help for performance and dependability analysis of IoT Cloud systems?

Thanks for your attention

Hong-Linh Truong
Faculty of Informatics, TU Wien
hong-linh.truong@tuwien.ac.at
http://www.infosys.tuwien.ac.at/staff/truong