Cloud Computing Patterns

Fundamentals to Design, Build, and Manage Cloud Applications

Tutorial at SummerSoC 2013 (1 July – 6 July, 2013, Hersonissos, Crete, Greece)

Christoph Fehling , Frank Leymann

Institute of Architecture of Application Systems

University of Stuttgart
Universitätsstr. 38
70569 Stuttgart
Germany

Phone +49-711-685-88 486
Fax +49-711-685-88 472
e-mail Fehling
@iaas.uni-stuttgart.de

Design Steps for Cloud Applications using Patterns

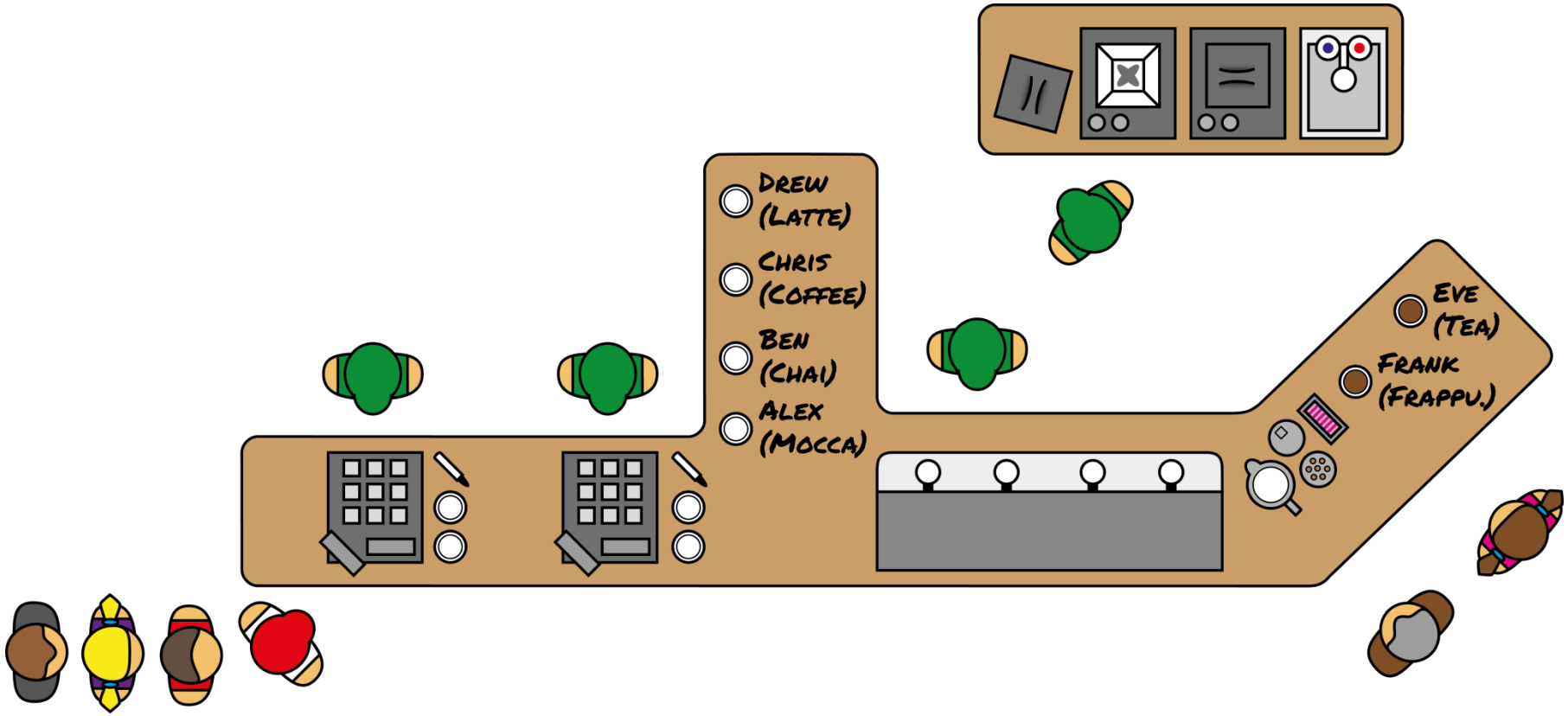
or

to see a Cloud Application Architecture you should go out and have a…

http://benedik.deviantart.com/

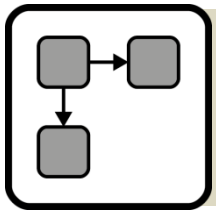
Coffee Shop

Decomposition → Workload → Data (State) → Component Refinement → Elasticity and Resiliency



Inspired by: G. Hohpe: Your Coffee Shop Doesn't Use Two-Phase Commit, IEEE Software, 2005.

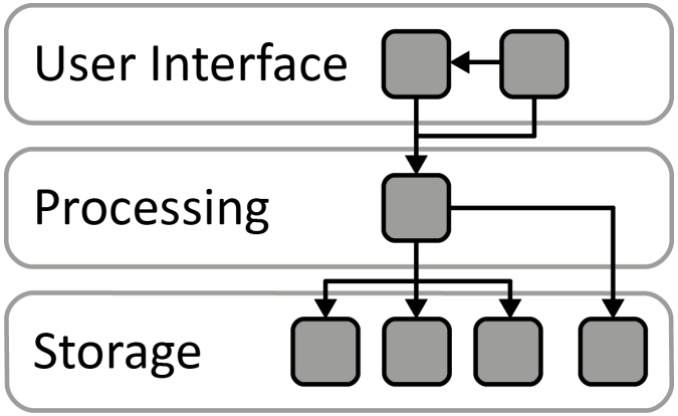
How to distribute Application Functionality?



Distributed Application

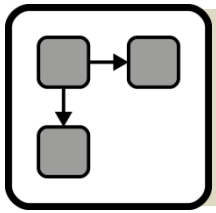
A cloud application divides provided functionality among multiple application components that can be scaled out independently.

Layer-based Decomposition



Components reside on separate functional layers

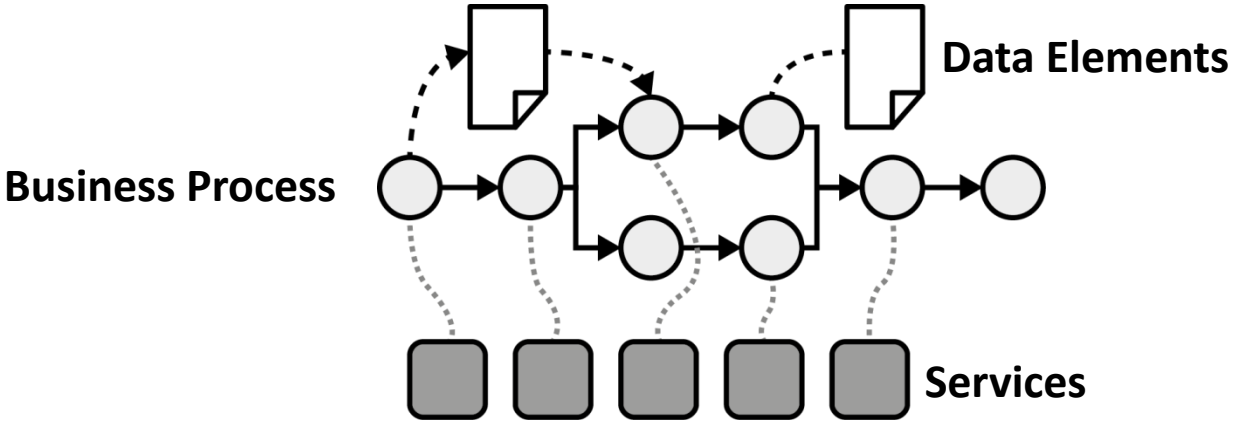
- Often: user interface, processing, storage
- Access is only allowed to **same layer and the layer below**
- →Dependencies between layers and interfaces are controlled



Distributed Application

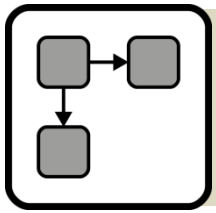
A cloud application divides provided functionality among multiple application components that can be scaled out independently.

Process-based Decomposition



Business process model determines decomposition

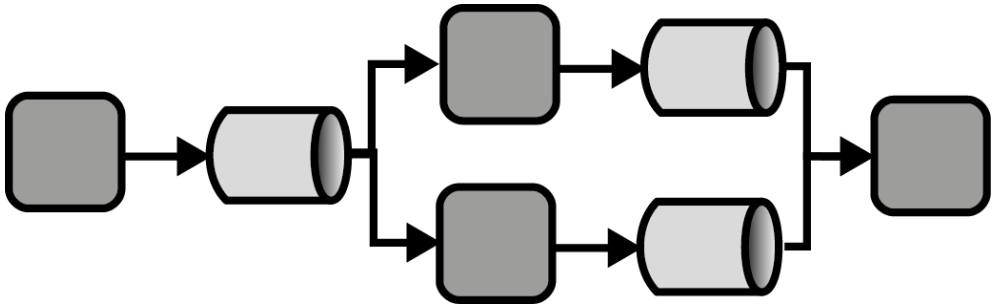
- **Activities**: tasks executed in a specific order (**control flow**)
- **Data elements**: information handled by activities (**data flow**)
- Functional application components (**services**) are accessed by process



Distributed Application

A cloud application divides provided functionality among multiple application components that can be scaled out independently.

Pipes-and-Filters-based Decomposition



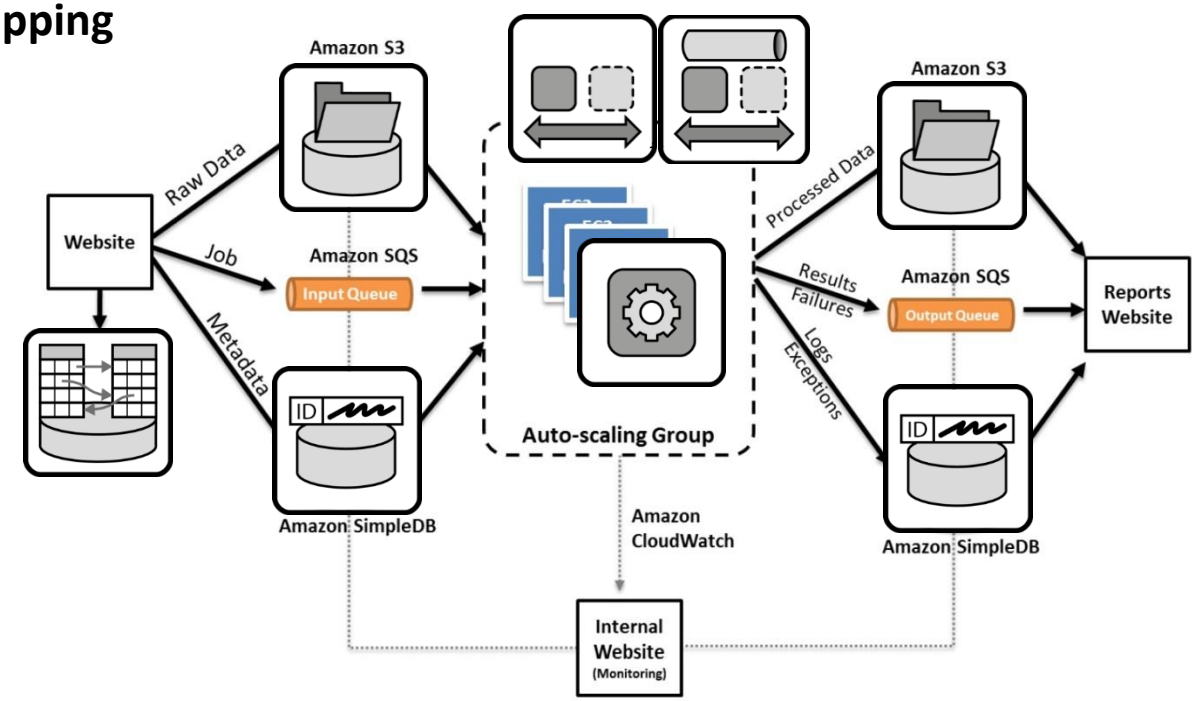
Decomposition based on the data processing function

- **Filter**: application component processing data
- **Pipe**: connection between filters (commonly messaging)

Have we seen this in the real world?

Yes: Migrating Batch Processes to the AWS Cloud

Pattern Mapping

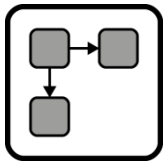


- RDBMS Data: can be handled by a *Relational Database*
- Amazon SimpleDB: *Key-value Storage*
- Amazon S3: *Blob Storage*
- Auto Scaling Group: *Elasticity Manager* or *Elastic Queue*
  (see Cloud Management Patterns)
- EC2 Worker Instance: *Processing Component*
  (*Batch Processing Component* if media is only processed when the queue is full)

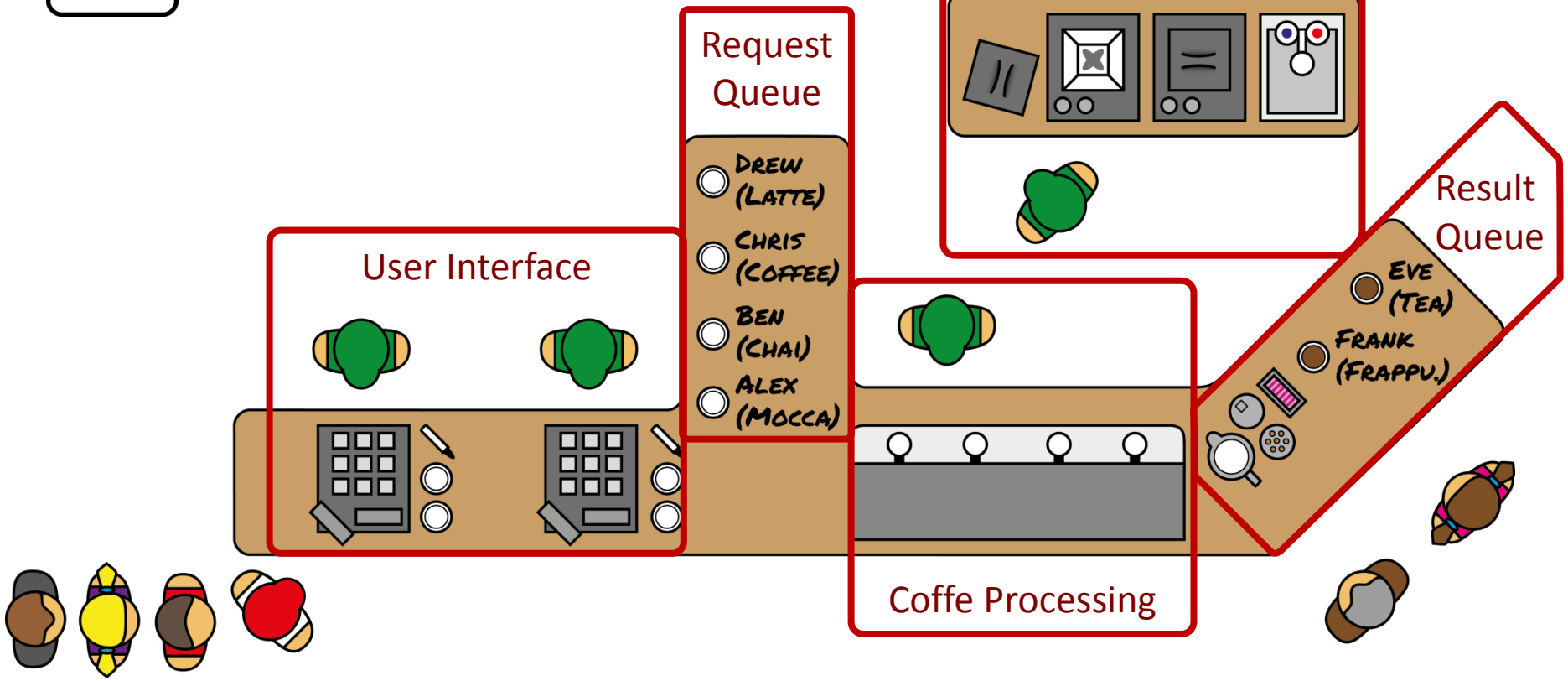
Source: Amazon Web Services White Paper: Migration Scenarios: Batch Processing

Coffe Shop – Decomposition of Functions

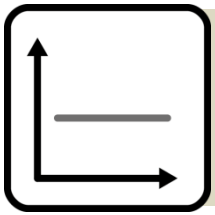
Identify functional components.



Distributed Application

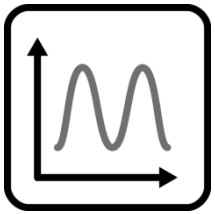


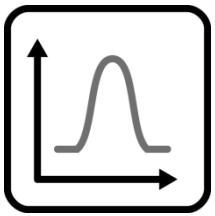
What workload do components experience?

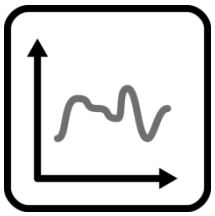


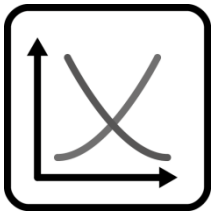
Static Workload

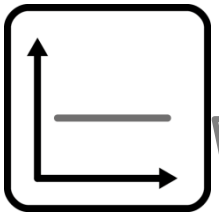
IT resources with an equal utilization over time experience static workload.

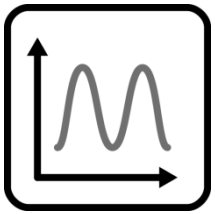


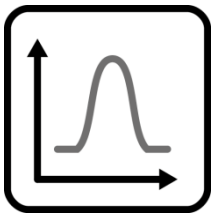


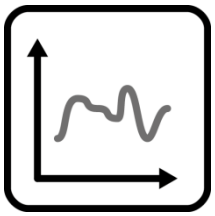


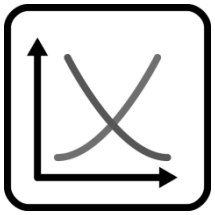




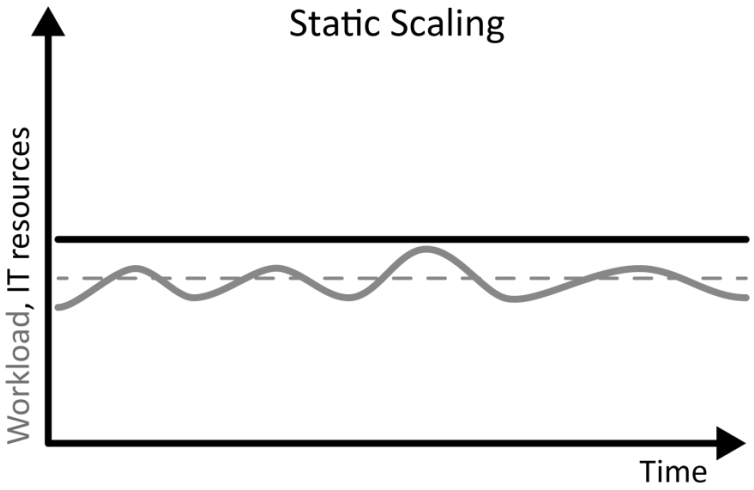


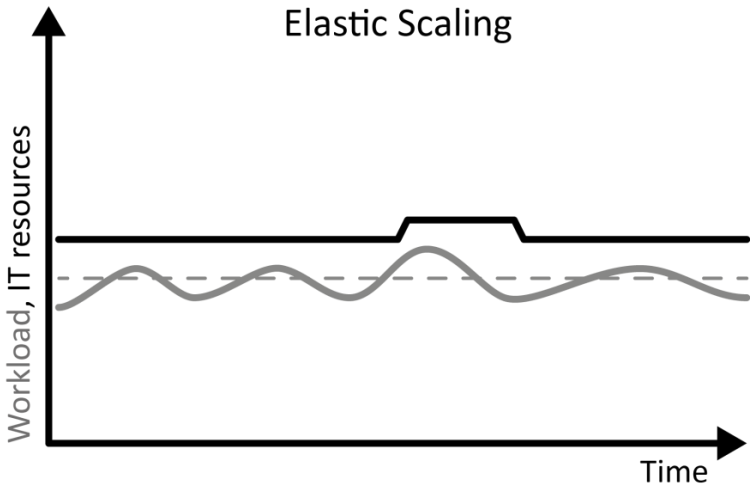






Static Workload





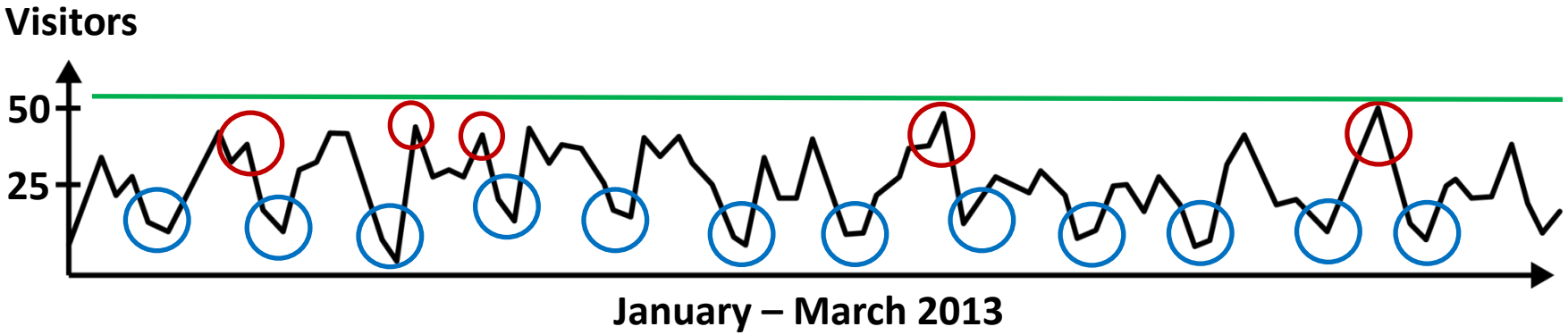
Predicted Workload

Experienced Workload

IT Resources

cloudcomputingpatterns.org

http://cloudcomputingpatterns.org



Low workload: every weekend

→ Page is obviously work-related

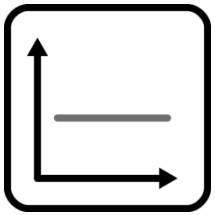
Peak workload: often at beginning / end of week

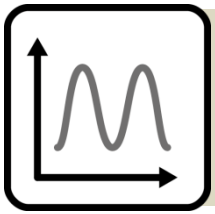
→ Probably a lot of meetings

Overall workload: does not utilize one server fully

→ **Hopefully this changes!**

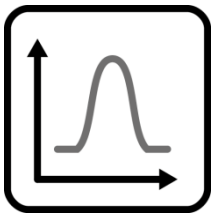
Source: http://www.google.com/analytics

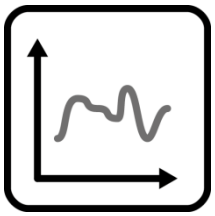


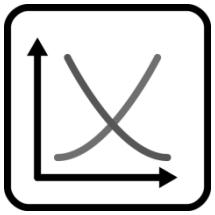


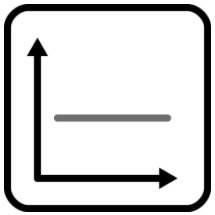
Periodic Workload

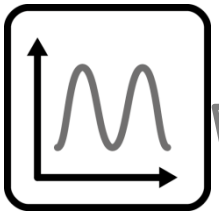
IT resources with a peaking utilization at reoccurring time intervals experience periodic workload.

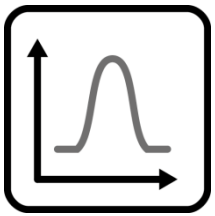


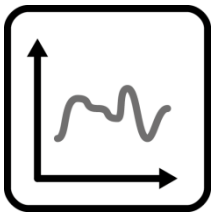


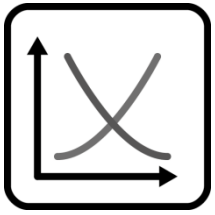




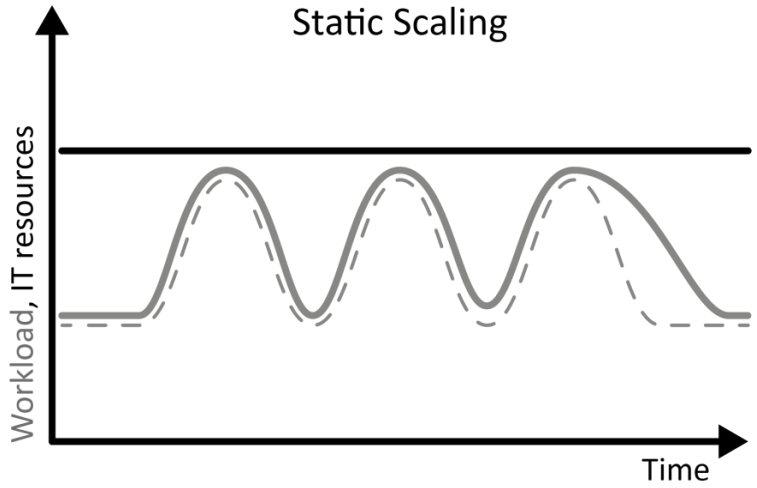


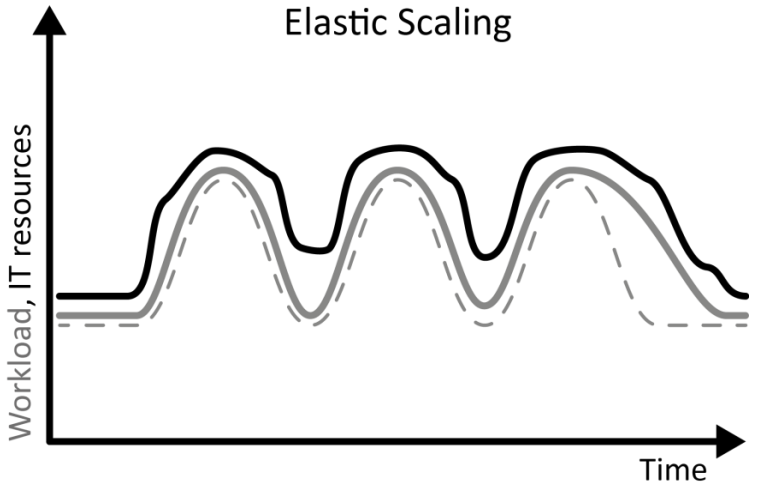






Periodic Workload





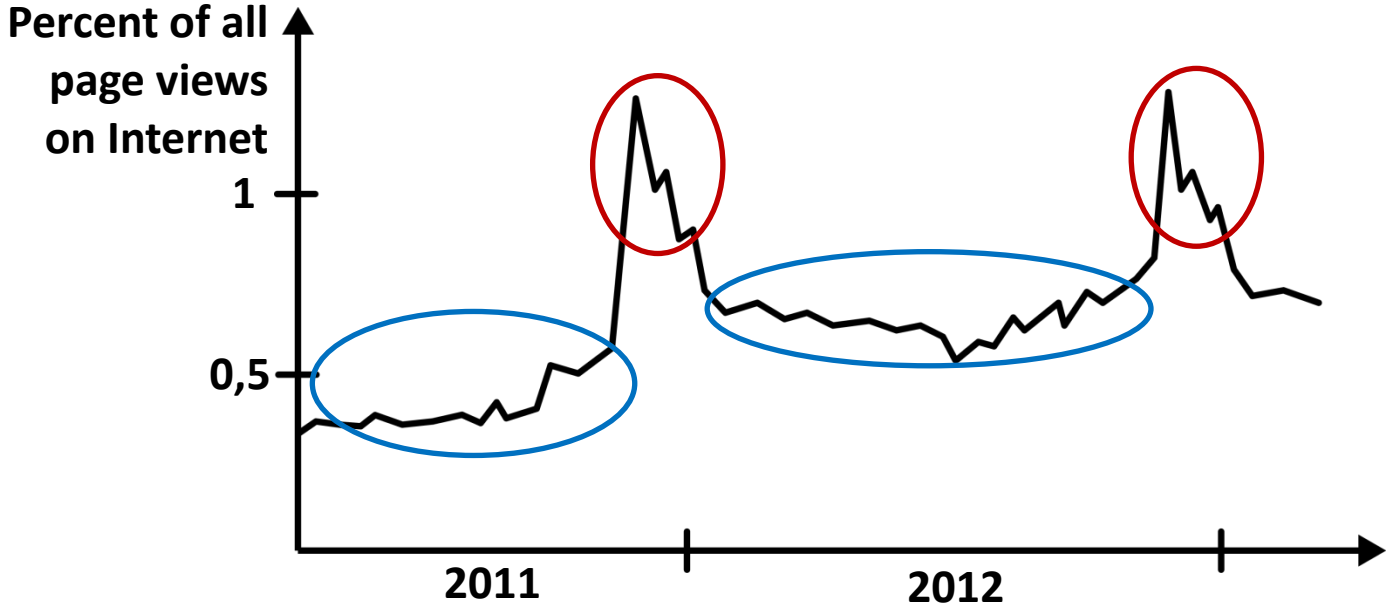
Predicted Workload

Experienced Workload

IT Resources

Amazon & YouTube

Amazon.com



Peak workload: very strong peaks at the end of every year

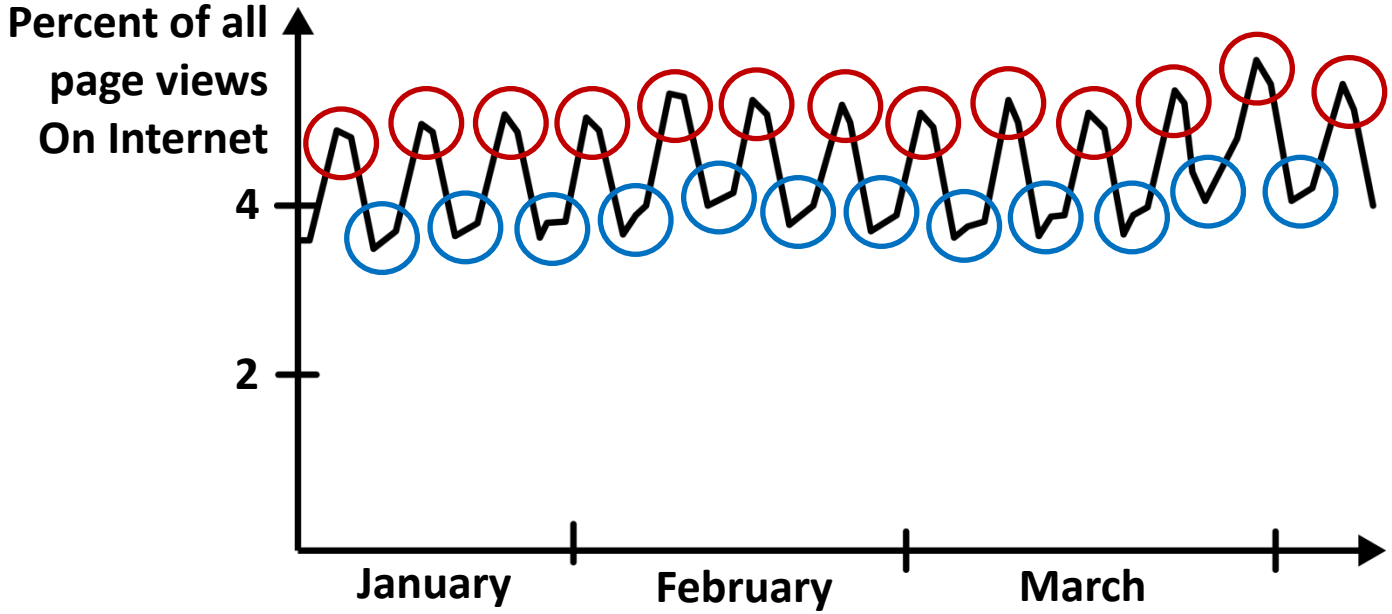
→ Christmas shopping

Low workload: no significant peaks during the year

→ Slight increase from year to year (we will cover such workload later)

Source: http://www.alexa.com

Youtube.com



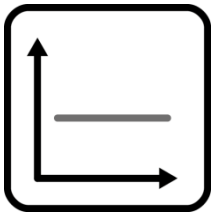
Peak workload: every weekend

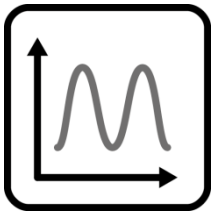
→ People are bored, listen to music etc.

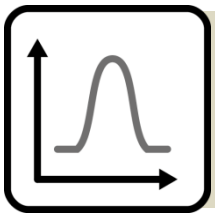
Low workload: during the week

→ People are at work

Source: http://www.alexa.com

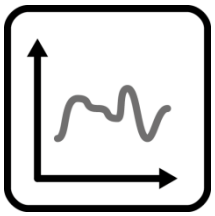


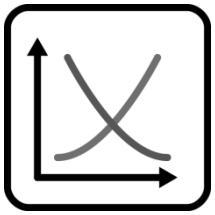


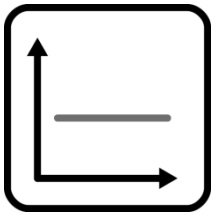


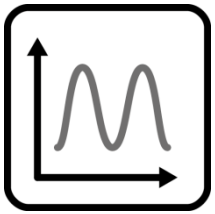
Once-in-a-Lifetime Workload

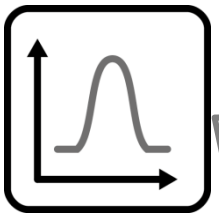
IT resources with an equal utilization over time disturbed by a strong peak occurring only once experience once-in-a-lifetime workload.

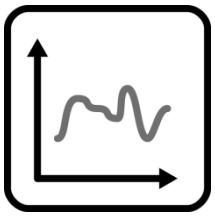


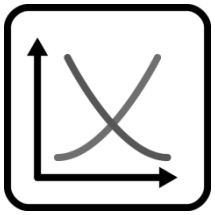


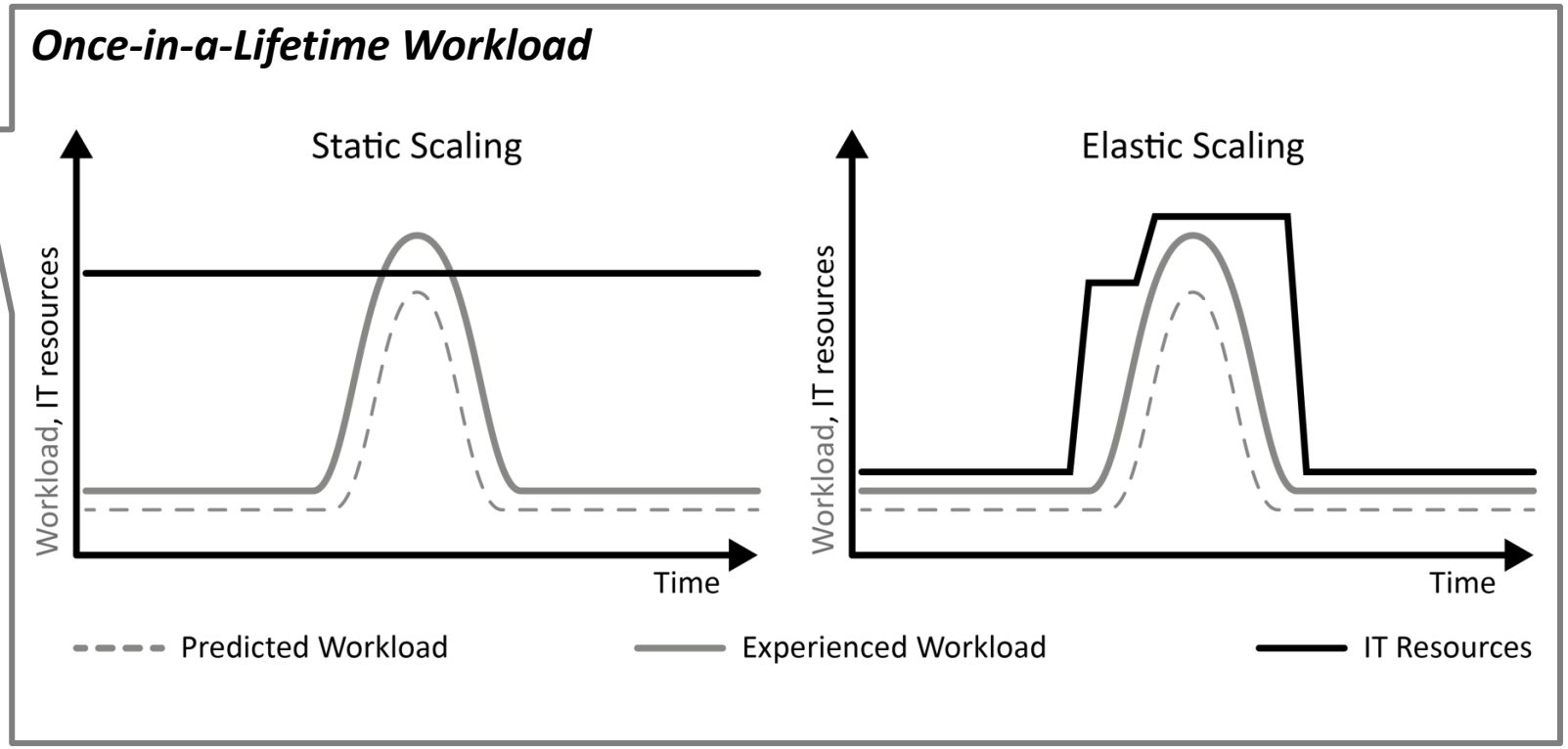











Once-in-a-Lifetime Workload
Static Scaling
Elastic Scaling
Workload, IT resources
Time
Workload, IT resources
Time
Predicted Workload
Experienced Workload
IT Resources

Digitalization of the New Your Times Archive

Digitalization of New York Times Archive

1 Original Data


The New York Times.

TITANIC SINKS FOUR HOURS AFTER HITTING ICEBERG; 866 RESCUED BY CARPATHIA, PROBABLY 1250 PERISH; ISMAY SAFE, MRS. ASTOR MAYBE, NOTED NAMES MISSING

The Lost Titanic Being Towed Out of Belfast Harbor.

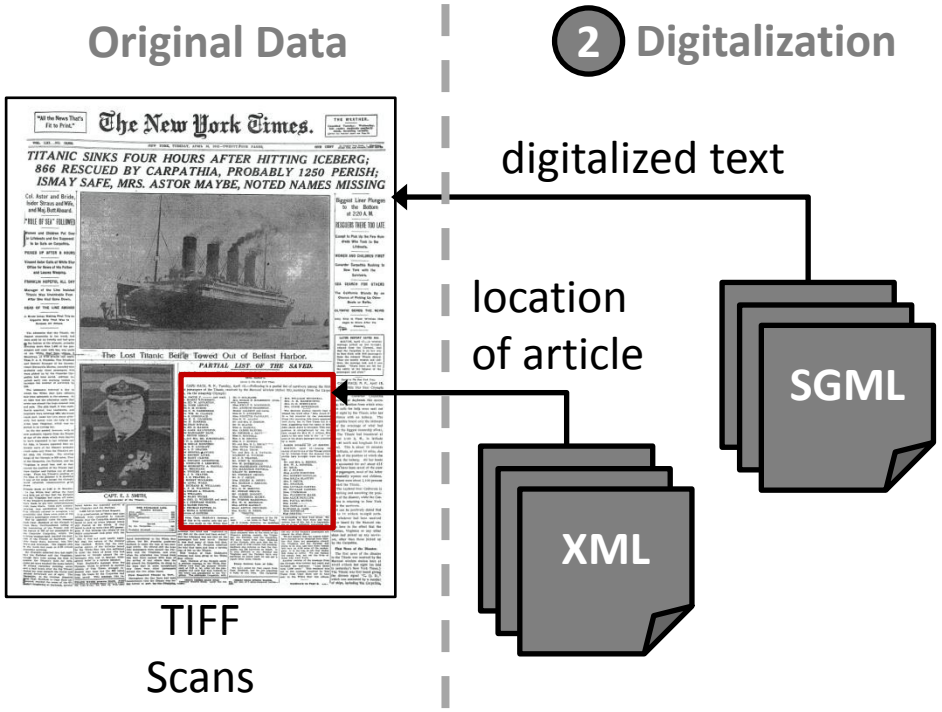
PARTIAL LIST OF THE SAVED.

TIFF
Scans

1 Original Data

- 150 years of news papers: 11 million articles
- Significant amount of public domain content
- Scanned to very large images in TIFF format (**4 TB**)

Digitalization of New York Times Archive



2 Digitalization: articles in PDF files with full text search

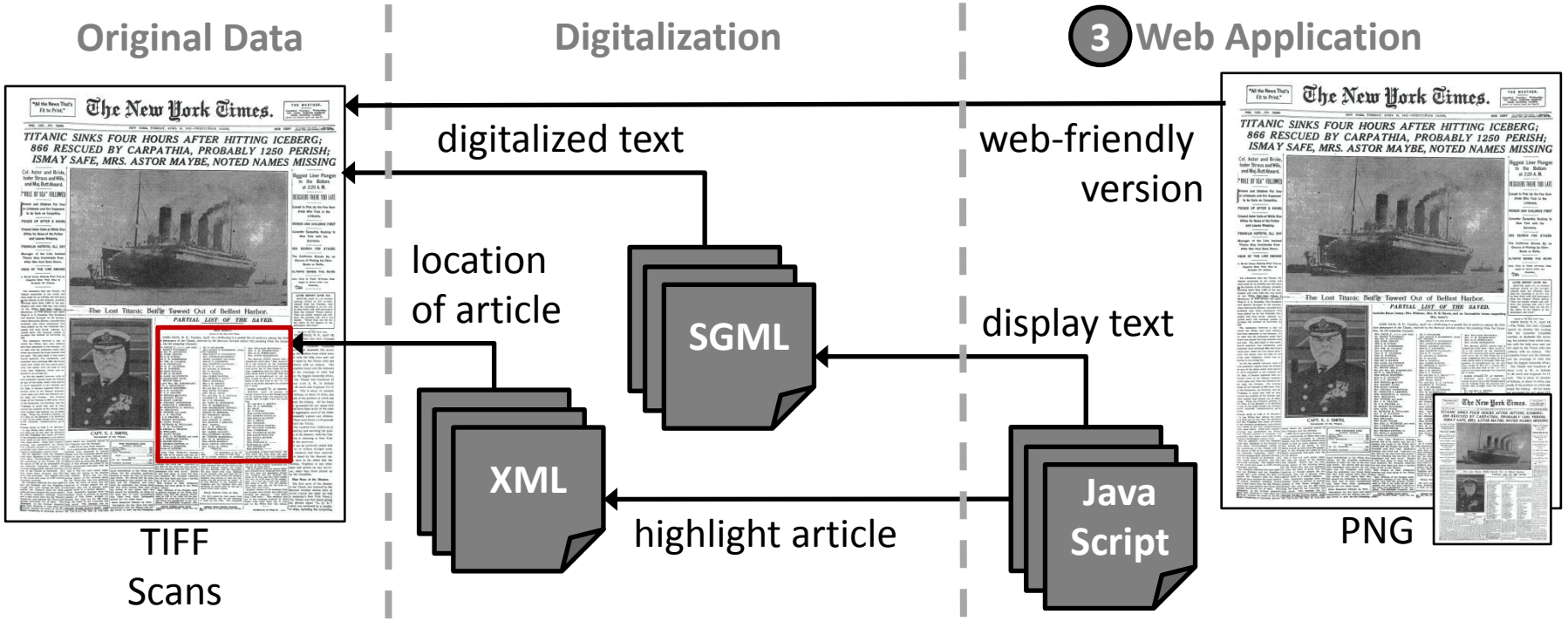
- XML (Extensible Markup Language): region (rectangle) of article
- SGML (Standard Generalized Markup Language): text of articles

→ Digitalization took **24 hours** on **100 Amazon EC2 machines**

→ Was **run twice** due to an error during the first run:

→ Output: **1,5 TB** of PDF files

Digitalization of New York Times Archive



3 Web Application: web-based access to 3.3 million articles

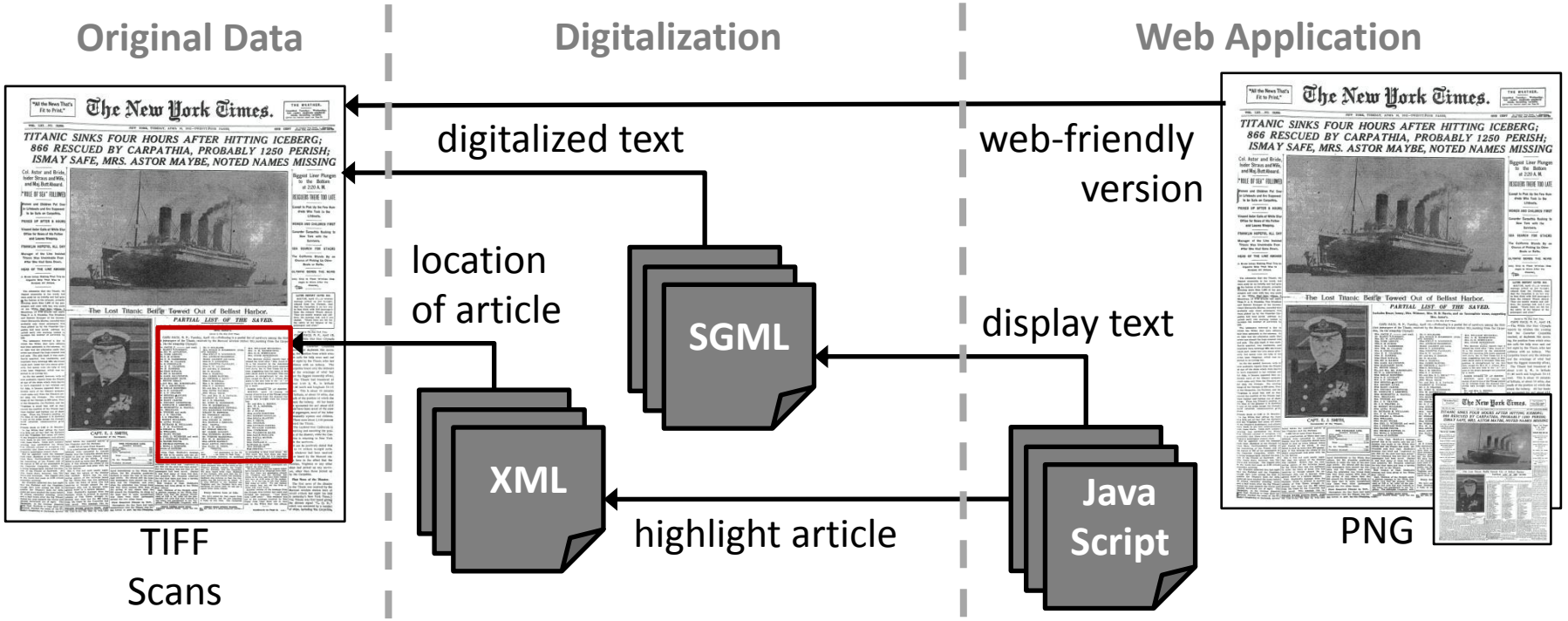
- XML +SGML: converted to Java Script handling the user interface
- TIFF files (405,000) converted to 810,000 PNG files (file + thumbnail)

→ Generation took less than 36 hours

→ Utilized „hundreds“ of Amazon EC2 machines

→ Output: foundation code for web application

Digitalization of New York Times Archive

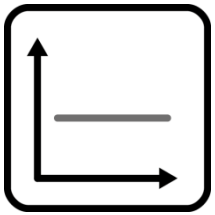


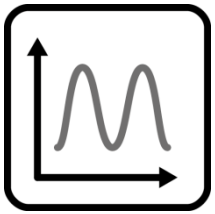
Sources

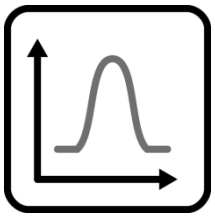
http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/

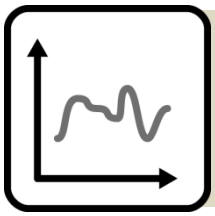
http://open.blogs.nytimes.com/2008/05/21/the-new-york-times-archives-amazon-web-services-timesmachine/

http://timesmachine.nytimes.com/



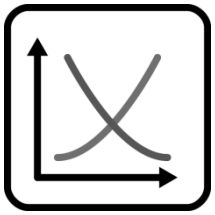


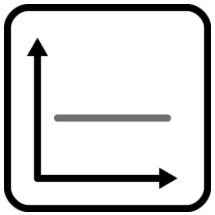


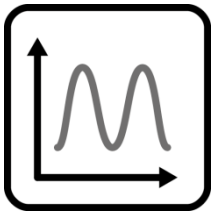


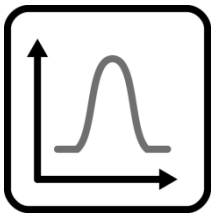
Unpredictable Workload

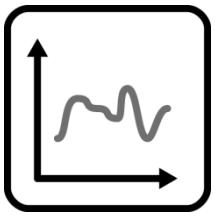
IT resources with a random and unforeseeable utilization over time experience unpredictable workload.

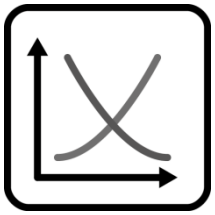




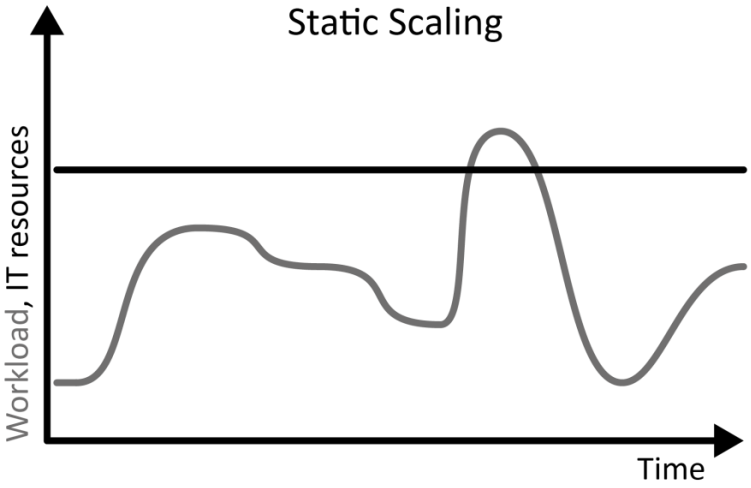


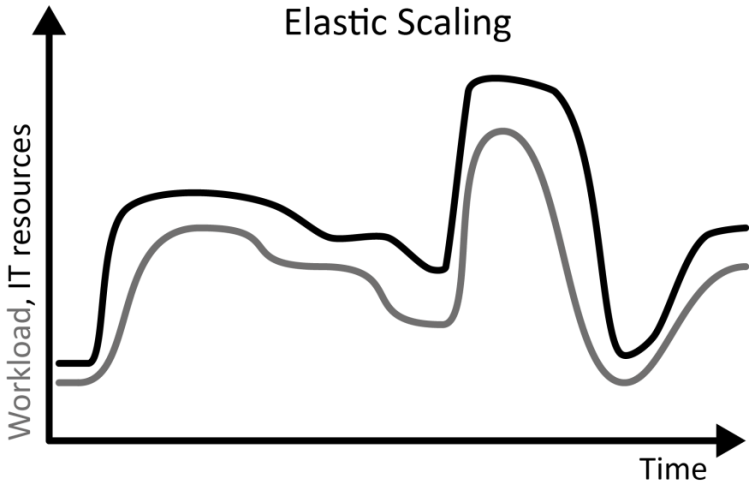






Unpredictable Workload



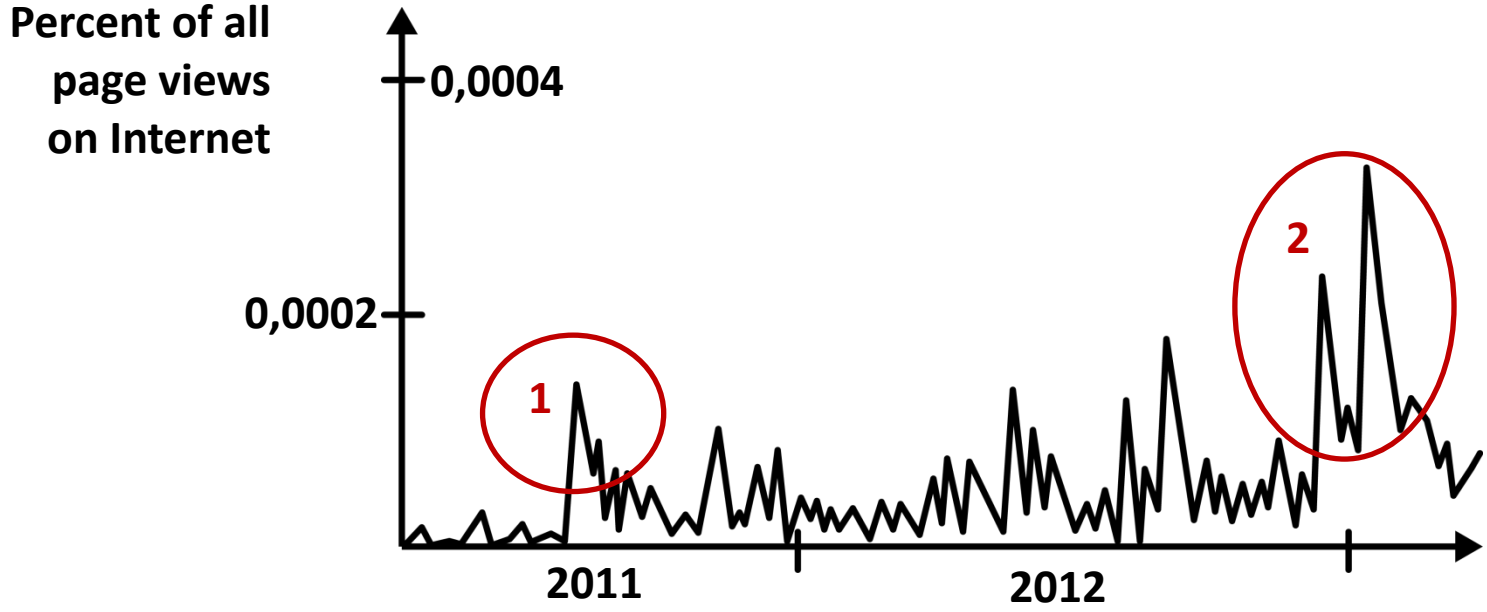


Experienced Workload

IT Resources

Eyeem.com Photo Sharing Website

Eyeem.com



1 **Peak workload October 2012**: High school Football Game in Nashville, TN

- One fan uploads pictures to Eyeem.com
- Sends picture on twitter

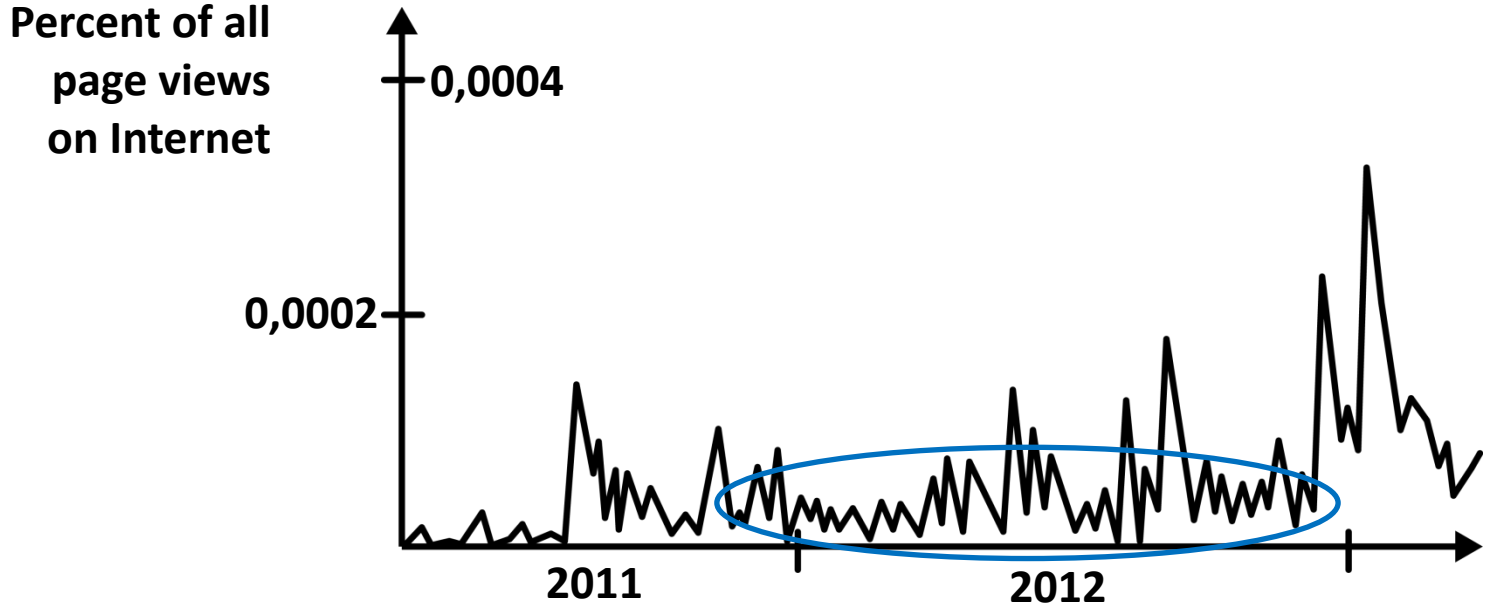
→ User group in Tennessee increases drastically!

2 **Peak workload October 2012**: Instagram changes privacy policy

- Instagram is similar picture sharing application
- Users are worried to loose rights to their images

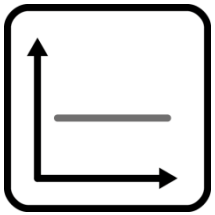
→ Massive migration of users to Eyeem.com

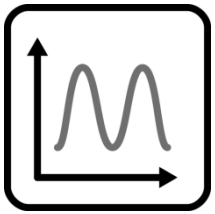
Eyeem.com

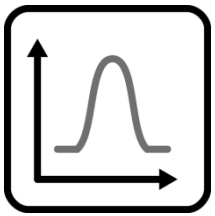


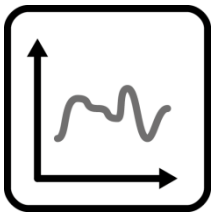
- **Low workload**: irregular and disturbed by small and sudden peaks
  - User access frequency changes often
  - Short peaks may be hard to detect without monitoring

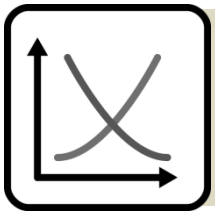
Source: http://www.alexa.com
http://www.golem.de/news/vom-erfolg-bedroht-der-kampf-ums-technische-ueberleben-1303-98323.html





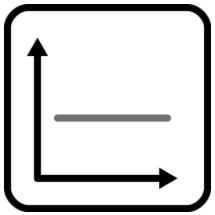


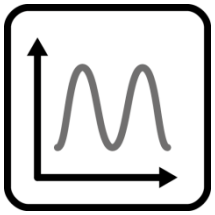


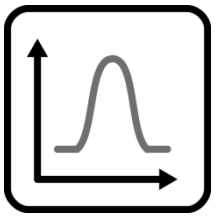


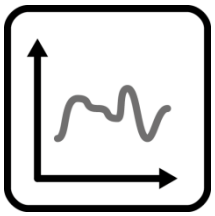
Continuously Changing Workload

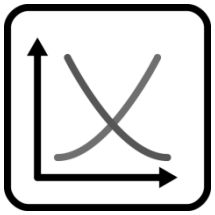
IT resources with a utilization that grows or shrinks constantly over time experience continuously changing workload.

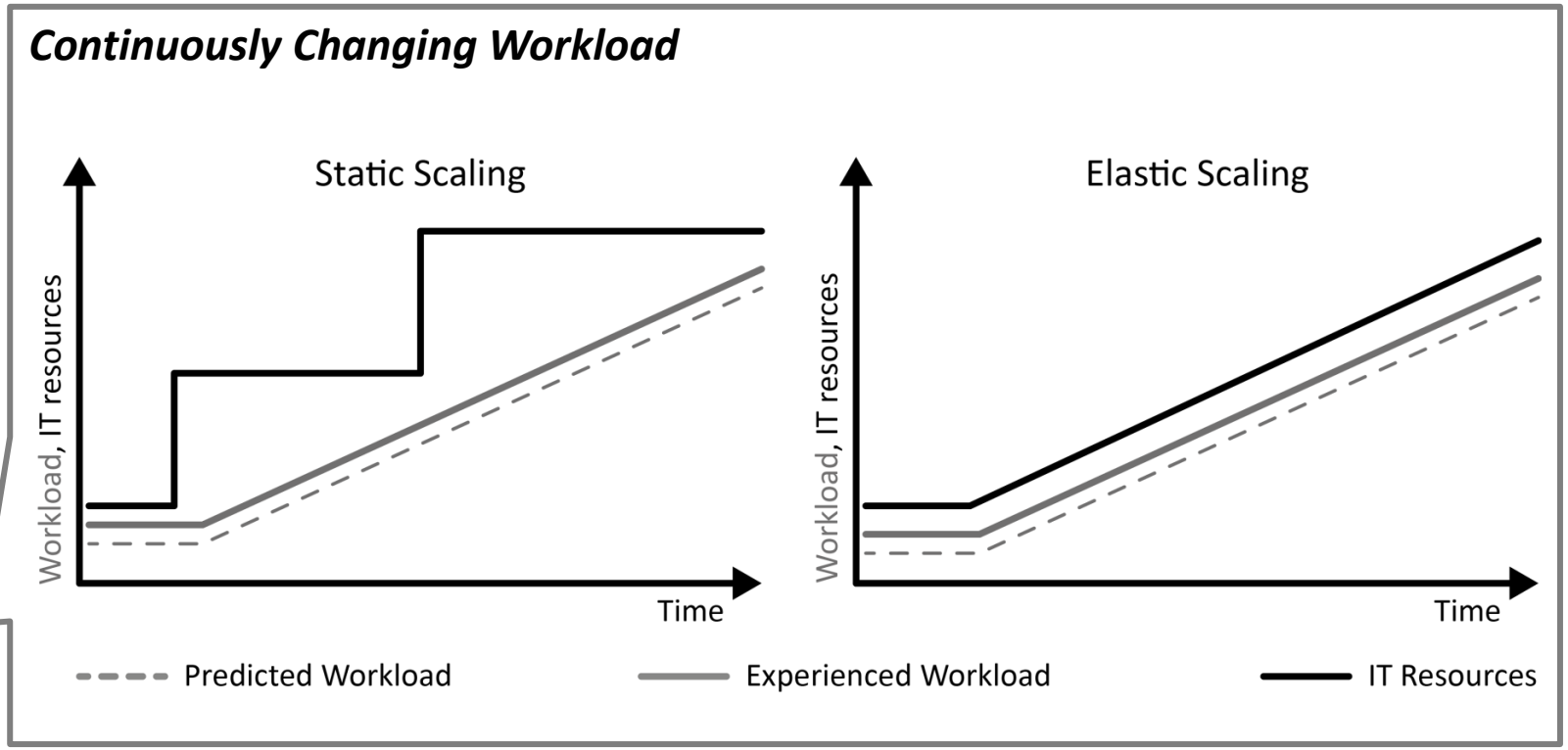






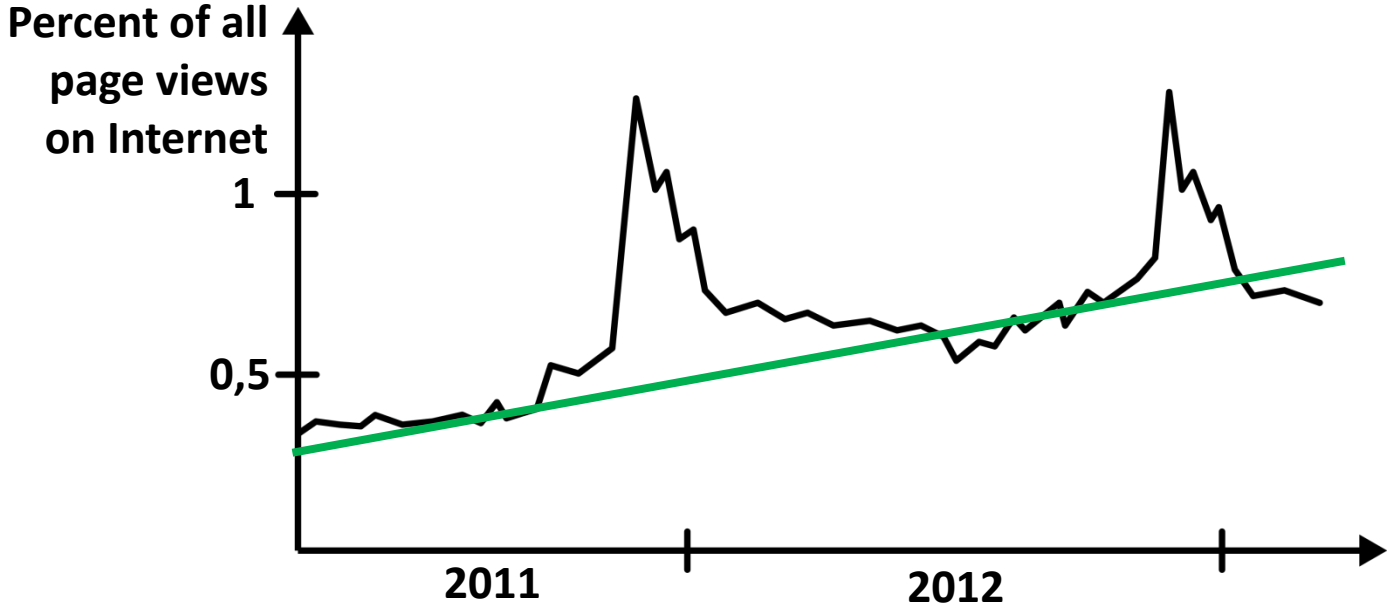





Continuously Changing Workload
Static Scaling
Elastic Scaling
Workload, IT resources
Time
Workload, IT resources
Time
Predicted Workload
Experienced Workload
IT Resources

Amazon

Amazon.com

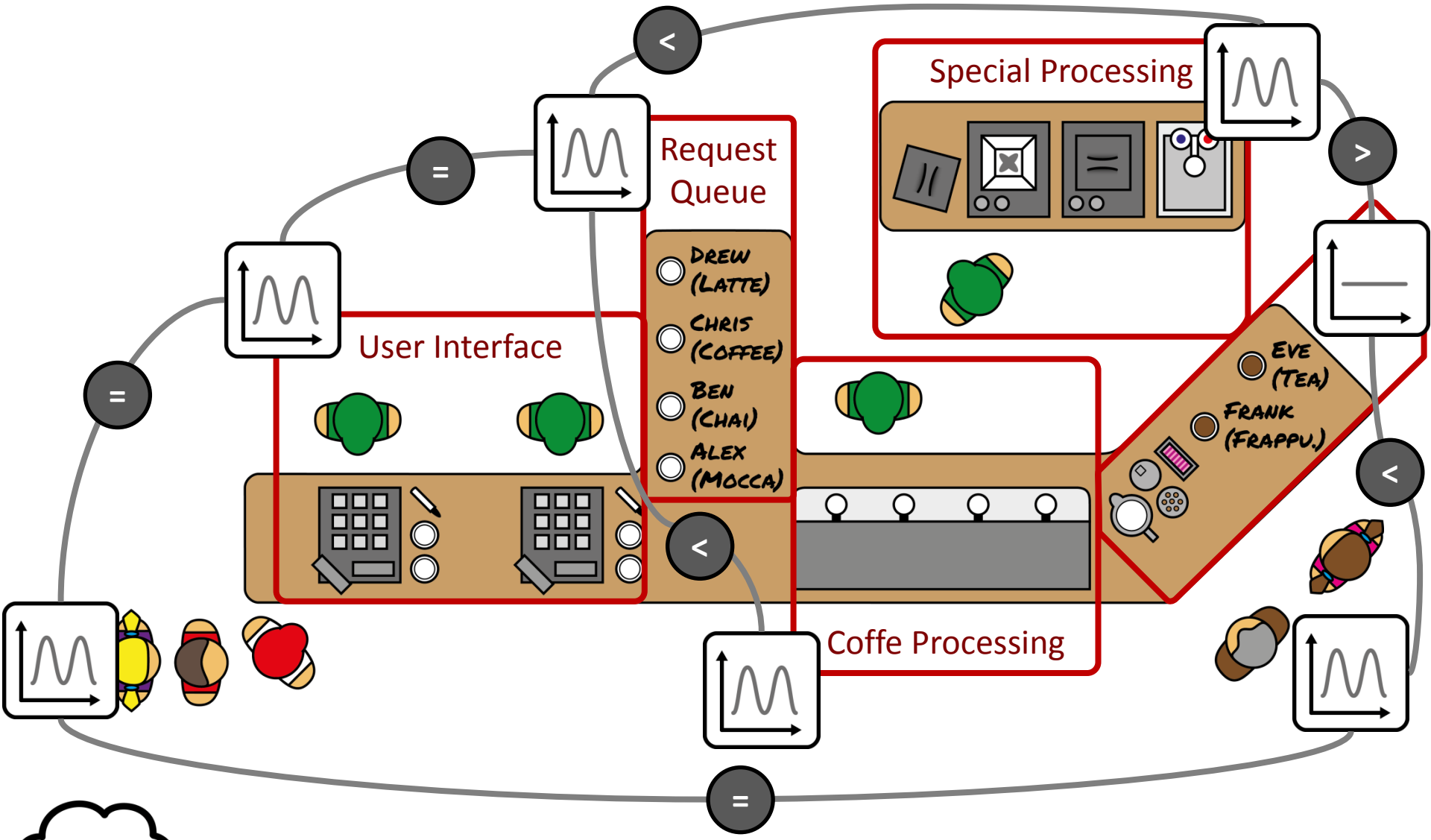


- **General growth**: between peaks workload increases continuously
  - → Amazon is growing not only during Christmas
  - → Different workload patterns can be combined!

Source: http://www.alexa.com

Coffe Shop – Workloads

Identify and compare workload generated by user groups at different components.



Lesson Learned

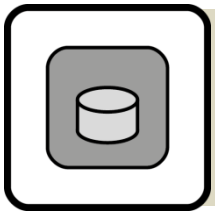
Workload can differ **significantly** between components.
Scaling them as a holistic unit can be very inefficient.

Where does the application handle state?

Notion of State

We differentiate between…

- **Session State**
  - *State of a client‘s interaction with an application*
  - Commonly referred to when discussing „statelessness“
  - Example: customer‘s shopping card of an online store
- **Application State**
  - Data handled by an application
  - We extend „statelessness“ to also incorporate application state
  - Example: customers shipping information stored by an application

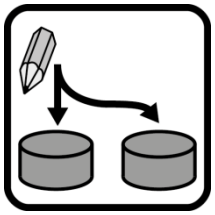


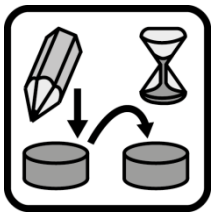
Stateful Component

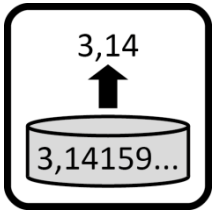
Multiple instances of a scaled-out application component synchronize their internal state to provide a unified behavior.

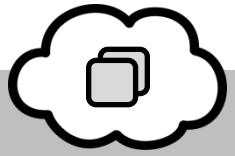
Stateless Component

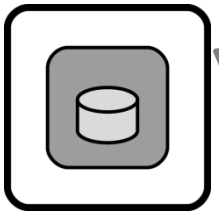
State is handled external of application components to ease their scaling-out and to make the application more tolerant to component failures.

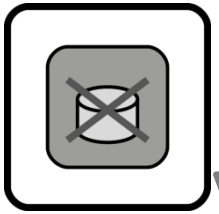


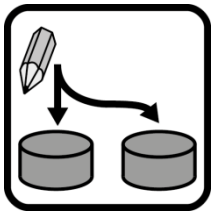


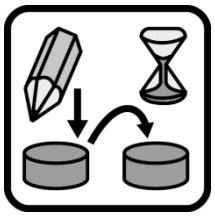


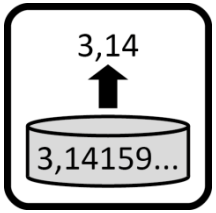


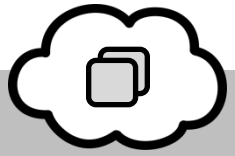


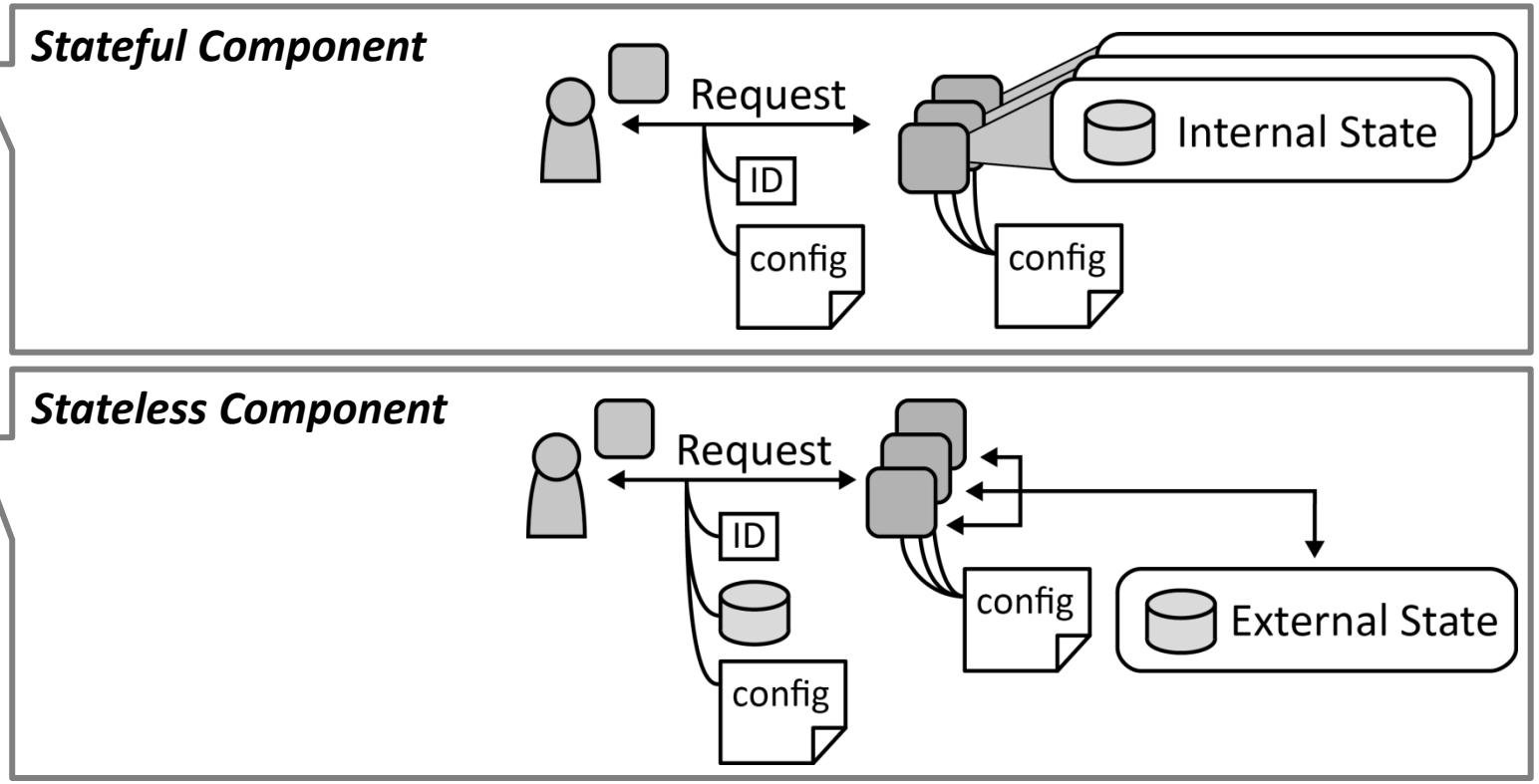


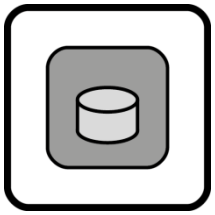


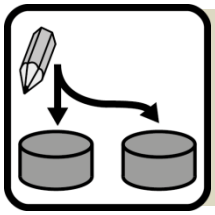



3,14
3,14159...



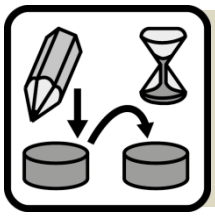

Stateful Component
Request
ID
config
Internal State
config
Stateless Component
Request
ID
config
config
External State





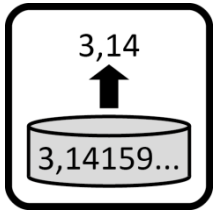
Strict Consistency

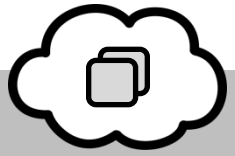
Data is stored at different locations to improve response time and to avoid data loss in case of failures while consistency of replicas is ensured at all times.

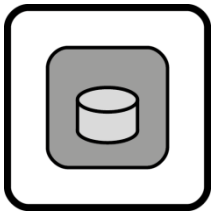


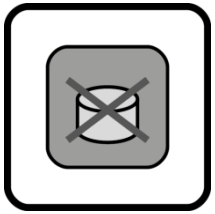
Eventual Consistency

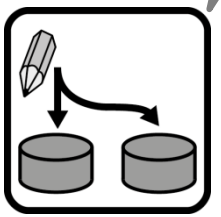
Performance and availability of data in case of network partitioning are enabled by ensuring data consistency eventually and not at all times.

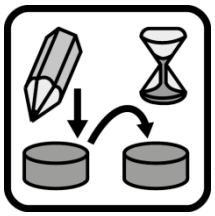


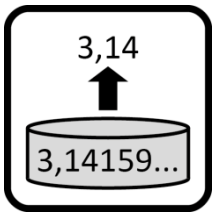




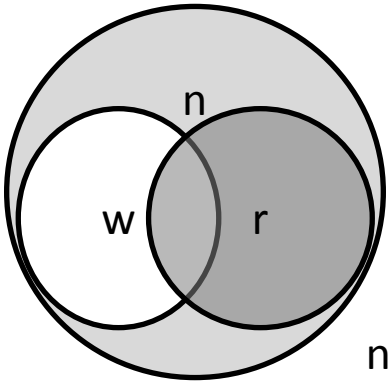




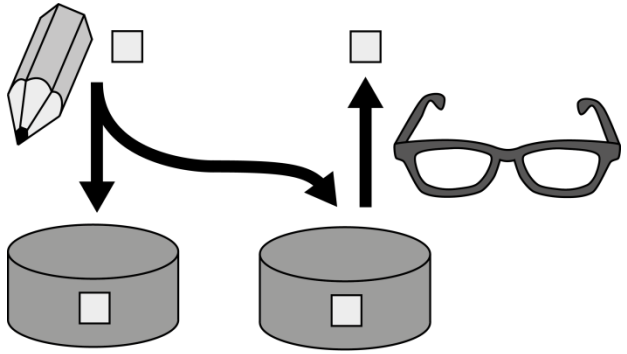




Strict Consistency

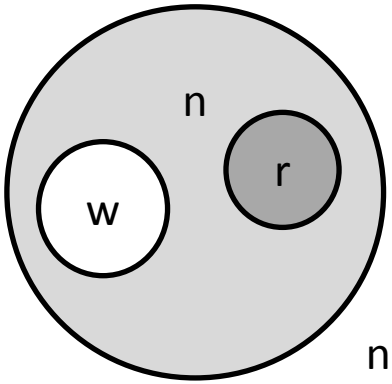


$n \leq r + w$

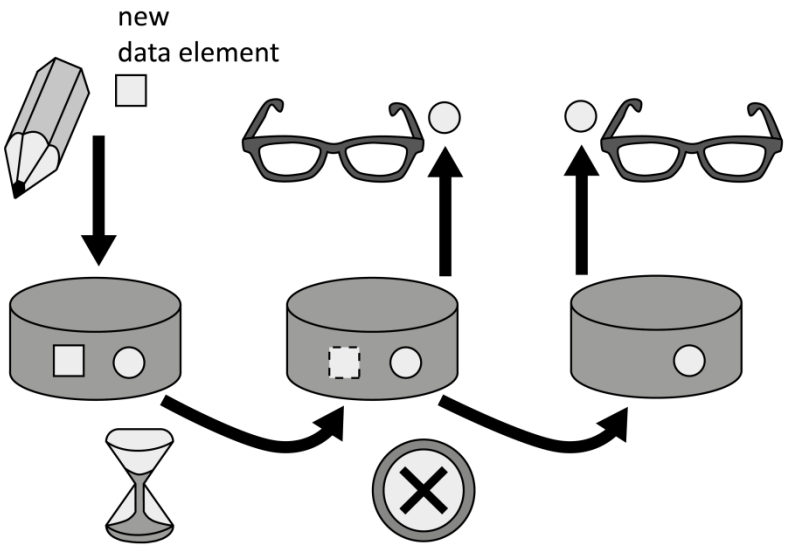


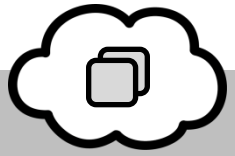
number of replicas (n) = 2
replicas accessed to write (w) = 2
replicas accessed to read (r) = 1

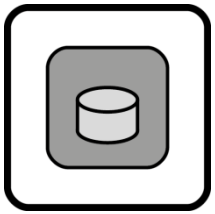
Eventual Consistency

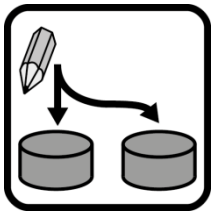


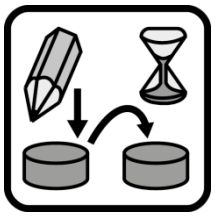
$n > r + w$

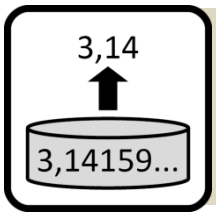






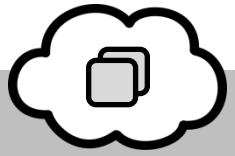


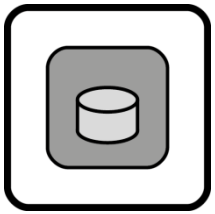


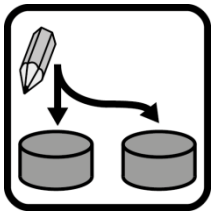


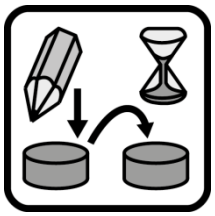
Data Abstractor

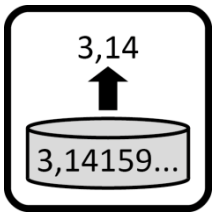
Data is abstracted to inherently support eventually consistent data storage through the use of abstractions and approximations.



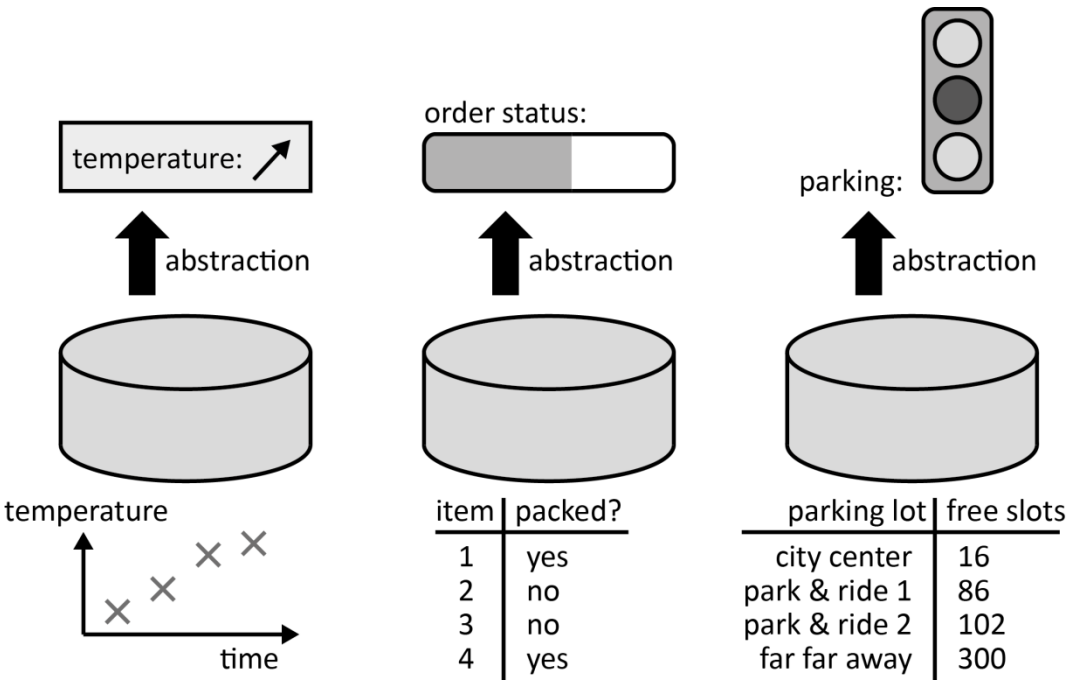


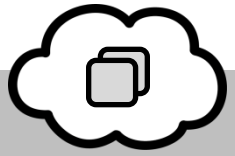






Data Abstractor



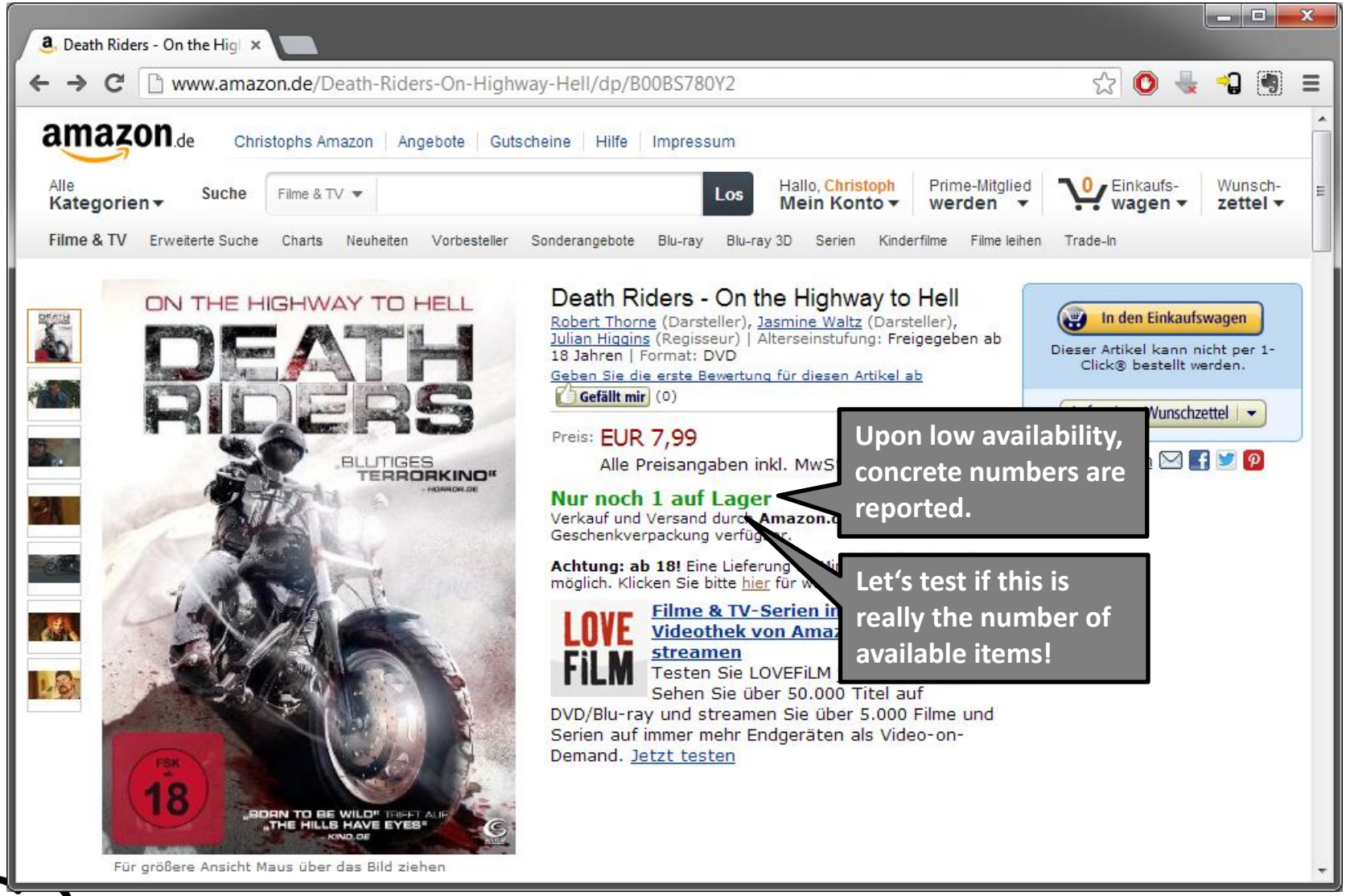


Data Abstraction at Amazon

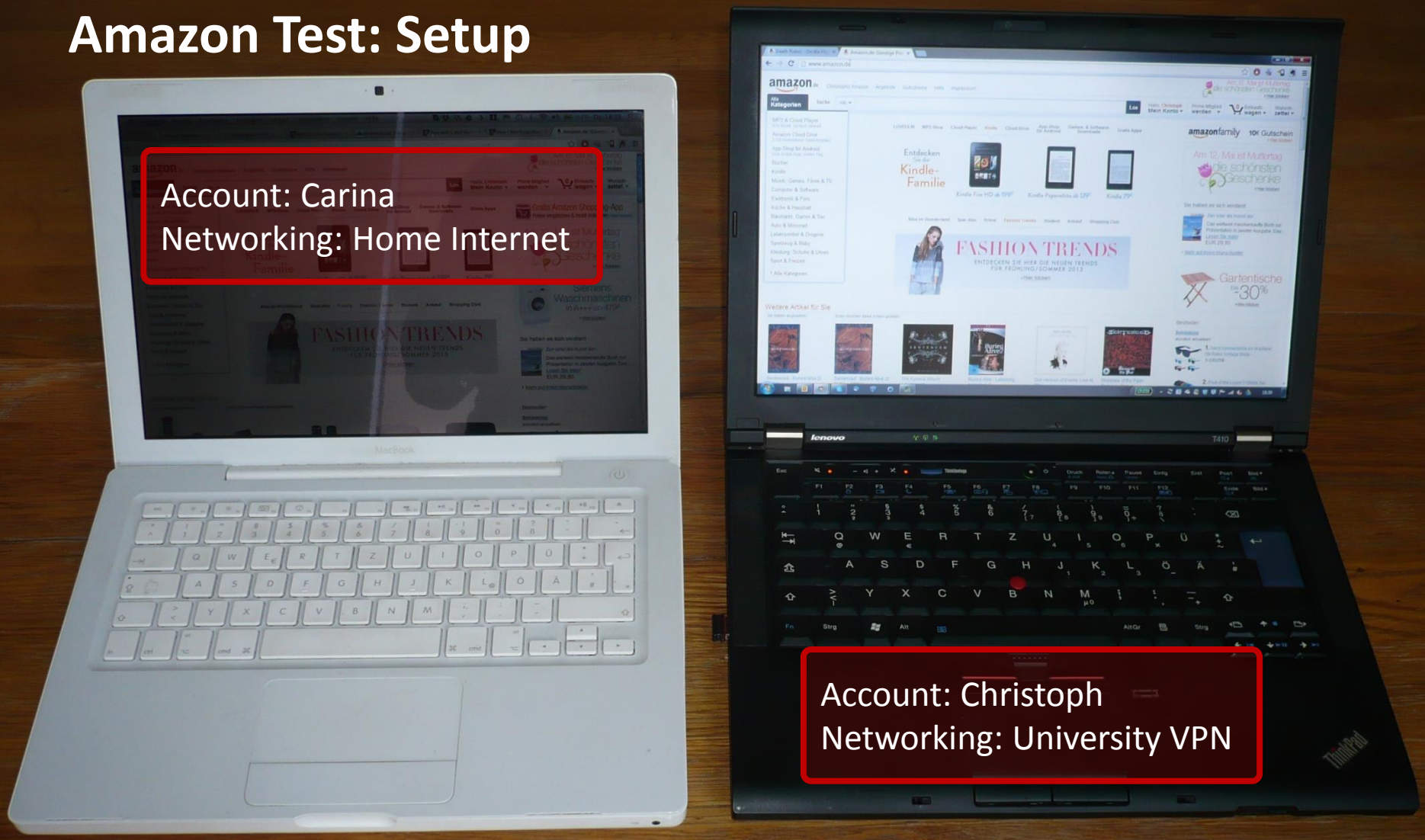
Amazon


Apple iMac MD096D/A 54
www.amazon.de/Apple-iMac-MD096D-Desktop-PC-Intel/dp/B00AEL5YBA/ref=sr_1_4?ie=UTF8&qid=1368182885&sr
amazon.de
Christophs Amazon
Angebote
Gutscheine
Verkaufen
Hilfe
Alle Kategorien
Suche
Computer & Zube...
imac
Los
Hallo, Christoph
Mein Konto
Prime-Mitglied werden
Einkaufswagen
Wunschzettel
Computer & Zubehör
Bestseller
Notebooks & PCs
Tablets
PC-Komponenten
PC-Zubehör & Monitore
Drucker & Tinte
Software
Sonderangebote
Apple iMac MD096D/A 54,6 cm (27 Zoll) Desktop-PC (Intel Core i5, 3,2GHz, 8GB RAM, 1TB HDD, Mac OS)
von Apple
(11 Kundenrezensionen)
Gefällt mir (10)
Preis: EUR 1.892,62 Kostenlose Lieferung. Details
Alle Preisangaben inkl. MwSt.
Auf Lager.
Verkauf und Versand durch Amazon.
Items are commonly „in stock“.
Stylisch-moderner All-in-One Desktopcomputer Hochleistungstechnologien innen und außen
Großes, brillantes, 5 mm dünnes LCD-Display mit LED-Hintergrundbeleuchtung reduziert Reflexionen um 75 Prozent und Sie genießen Ihre Medien extrem detailliert, lebendig und realistisch
Leistungsstark: Intel Core Prozessor der dritten Generation, NVIDIA Grafikprozessor, Thunderbolt und USB 3.0 -Anschlüsse, Fusion Drive Speichertechnologie, 720p FaceTime HD Kamera
NVIDIA GeForce Prozessor bietet bis zu 60 Prozent schnellere Grafik-Performance für grafikintensive Aufgaben
Lieferumfang: Apple iMac Desktopcomputer, Mac OS-Betriebssystem, Apple Wireless Keyboard, Apple Magic Mouse, Netzkabel, umfangreiche Software
Weitere Produktdetails
32 neu ab EUR 1.892,62
2 gebraucht ab EUR 1.780,01
2 Kundenbilder anzeigen
Für Kunden: Stellen Sie Ihre eigenen Bilder ein.
Menge: 1
In den Einkaufswagen
oder
Loggen Sie sich ein, um 1-Click® einzuschalten.
Notebook.de
In den Einkaufswagen
EUR 2.000,00 + kostenlose Lieferung.
ELECTRONIS
In den Einkaufswagen
EUR 2.000,00 + kostenlose Lieferung.
34 Angebote ab EUR 1.780,01
Empfehlen

Amazon

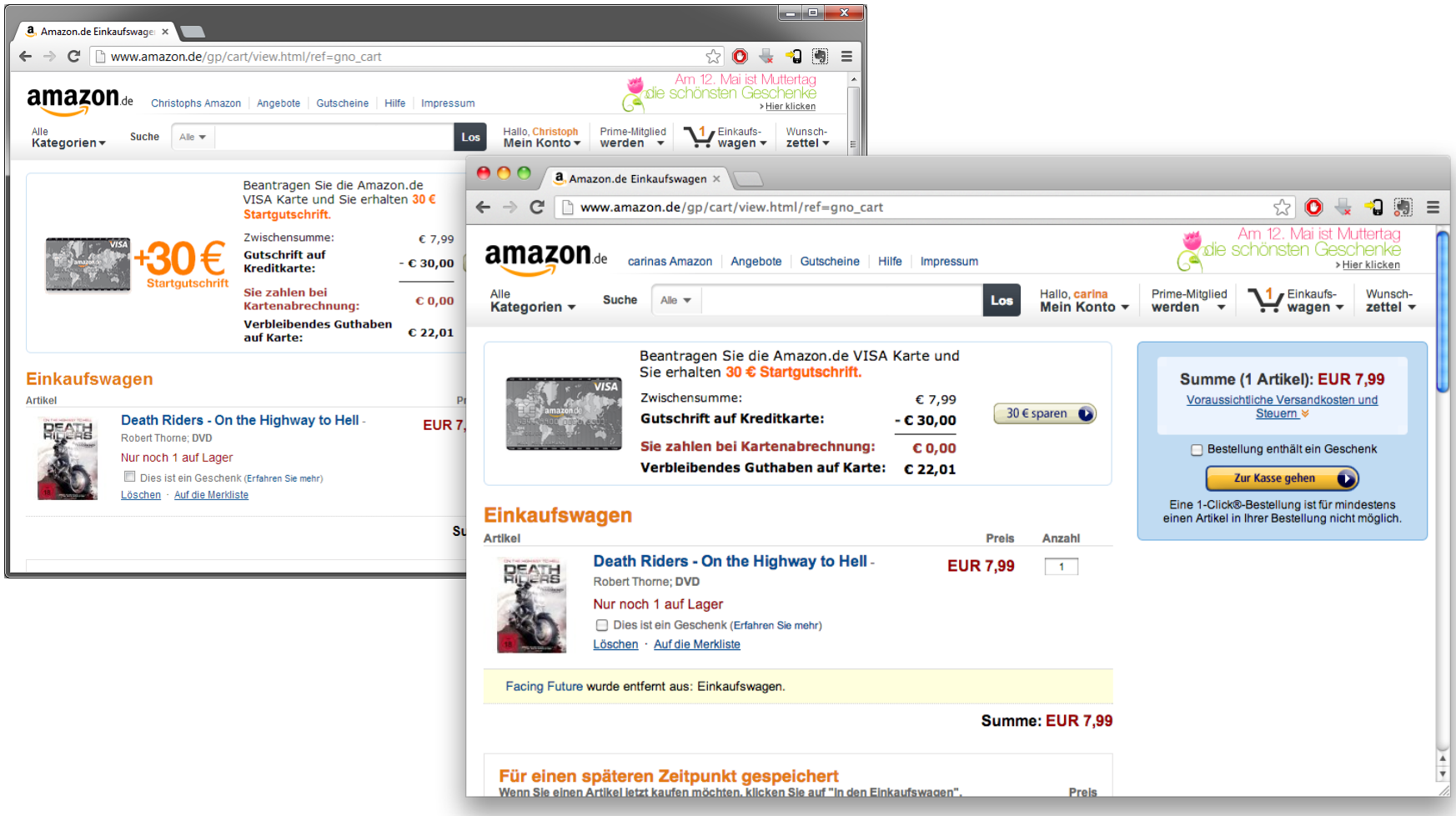

Death Riders - On the Hig
www.amazon.de/Death-Riders-On-Highway-Hell/dp/B00BS780Y2
amazon.de
Christophs Amazon | Angebote | Gutscheine | Hilfe | Impressum
Alle Kategorien
Suche
Filme & TV
Los
Hallo, Christoph Mein Konto
Prime-Mitglied werden
0 Einkaufswagen
Wunschzettel
Filme & TV | Erweiterte Suche | Charts | Neuheiten | Vorbesteller | Sonderangebote | Blu-ray | Blu-ray 3D | Serien | Kinderfilme | Filme leihen | Trade-In
ON THE HIGHWAY TO HELL
DEATH RIDERS
„BLUTIGES TERRORKINO"
FSK 18
Für größere Ansicht Maus über das Bild ziehen
Death Riders - On the Highway to Hell
Robert Thorne (Darsteller), Jasmine Waltz (Darsteller), Julian Higgins (Regisseur) | Alterseinstufung: Freigegeben ab 18 Jahren | Format: DVD
Geben Sie die erste Bewertung für diesen Artikel ab
Gefällt mir (0)
Preis: EUR 7,99
Alle Preisangaben inkl. MwS
Nur noch 1 auf Lager
Verkauf und Versand durch Amazon.
Geschenkverpackung verfügbar.
Achtung: ab 18! Eine Lieferung
möglich. Klicken Sie bitte hier für
In den Einkaufswagen
Dieser Artikel kann nicht per 1-Click® bestellt werden.
Wunschzettel
LOVE FiLM
Filme & TV-Serien in
Videothek von Amaz
streamen
Testen Sie LOVEFiLM
Sehen Sie über 50.000 Titel auf DVD/Blu-ray und streamen Sie über 5.000 Filme und Serien auf immer mehr Endgeräten als Video-on-Demand. Jetzt testen
Upon low availability, concrete numbers are reported.
Let's test if this is really the number of available items!

Amazon Test: Setup



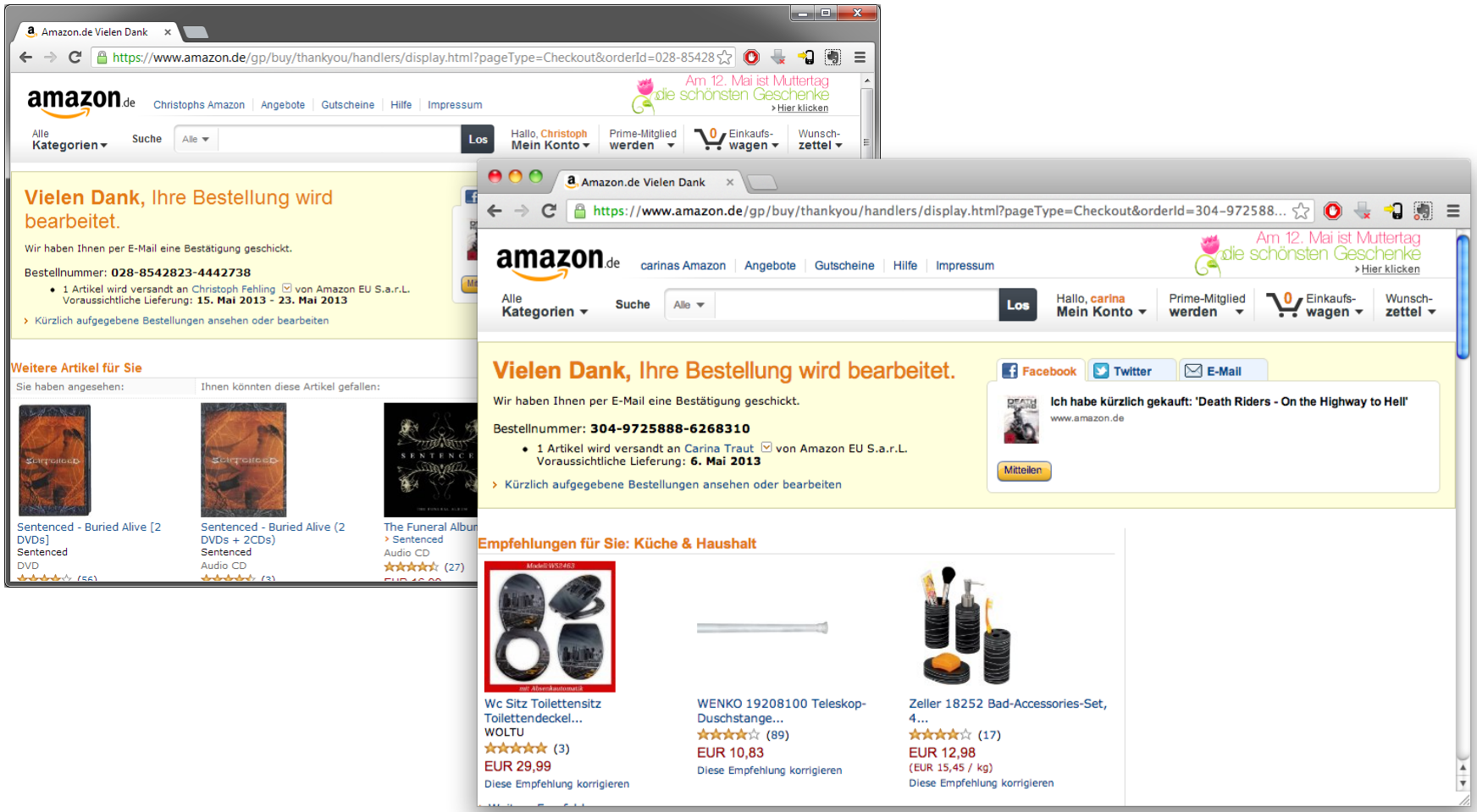
→Simulation of two different customers shopping at two different locations.

Amazon: Shopping Carts



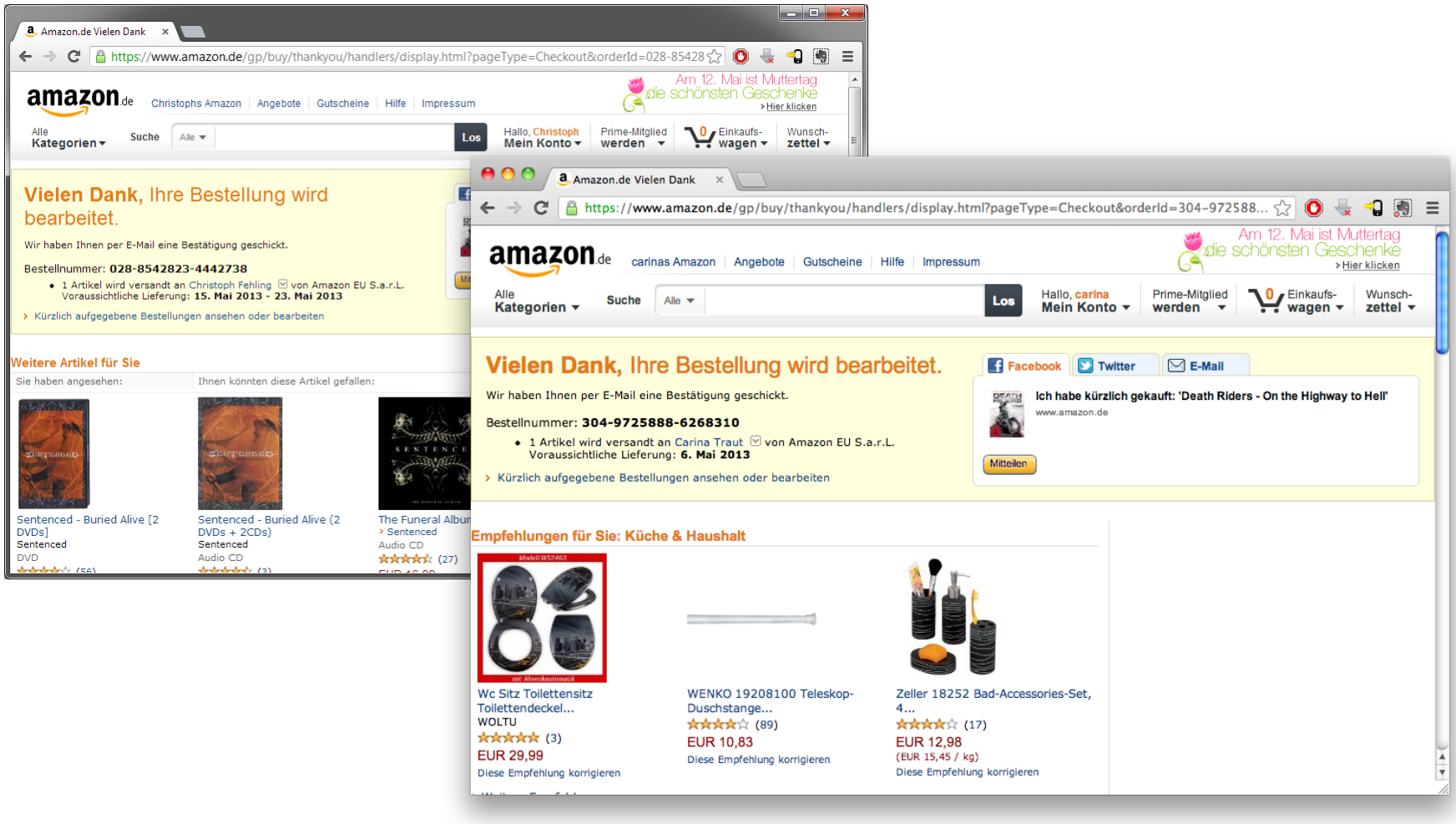
Item could be added to both shopping carts.

Amazon: Shopping Carts



Item could be bought by both customers.

Amazon: Shopping Carts



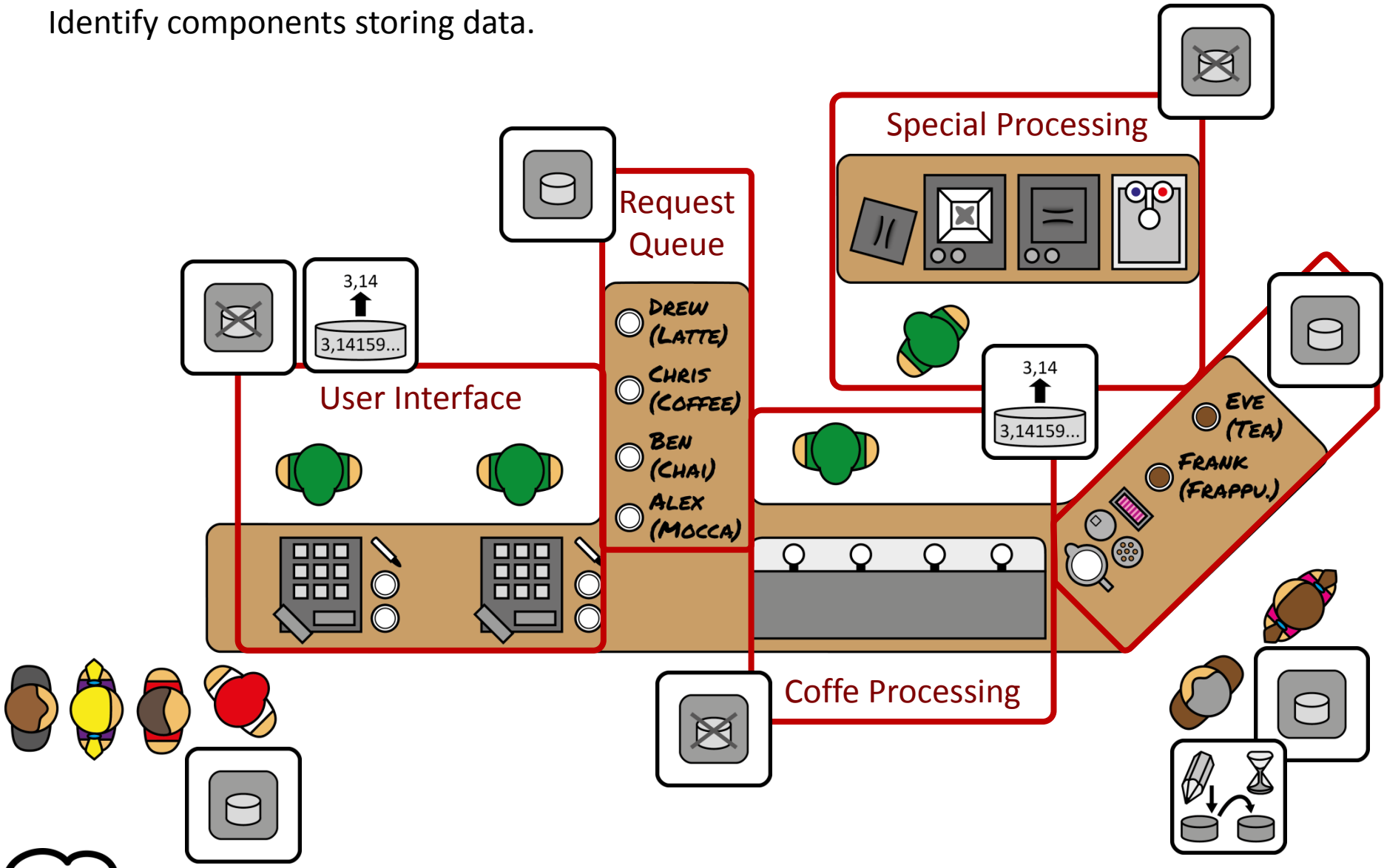
Item could be bought by both customers.

Amazon: Delivery

- Both customers received the item at the same day!
- The movie was horrible!!!
- Amazon Website one day after the delivery:

Coffee Shop – Data

Identify components storing data.



Lessons Learned

Avoid state in application components.

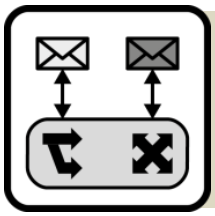
Handle state in…

… requests (has to be provided with every access).

… provider-supplied storage and communication offerings.

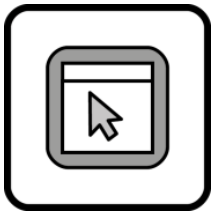
„**Lie**“ about state whenever possible / acceptable by the business case.

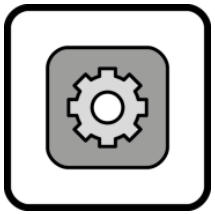
How are components implemented?

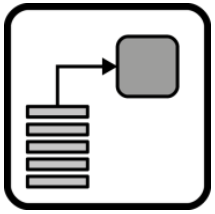


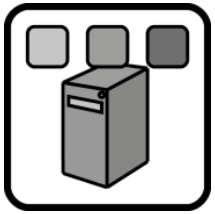
Message-oriented Middleware

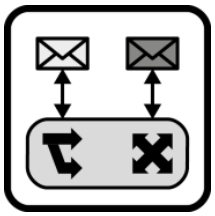
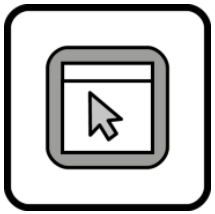
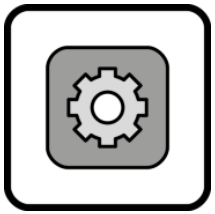
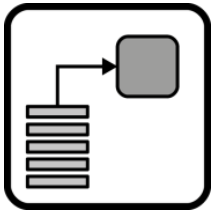
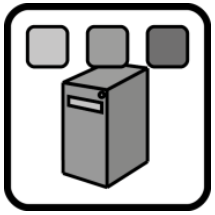
Asynchronous communication is provided while hiding complexity of addressing, routing, or data formats to make interaction robust and flexible.



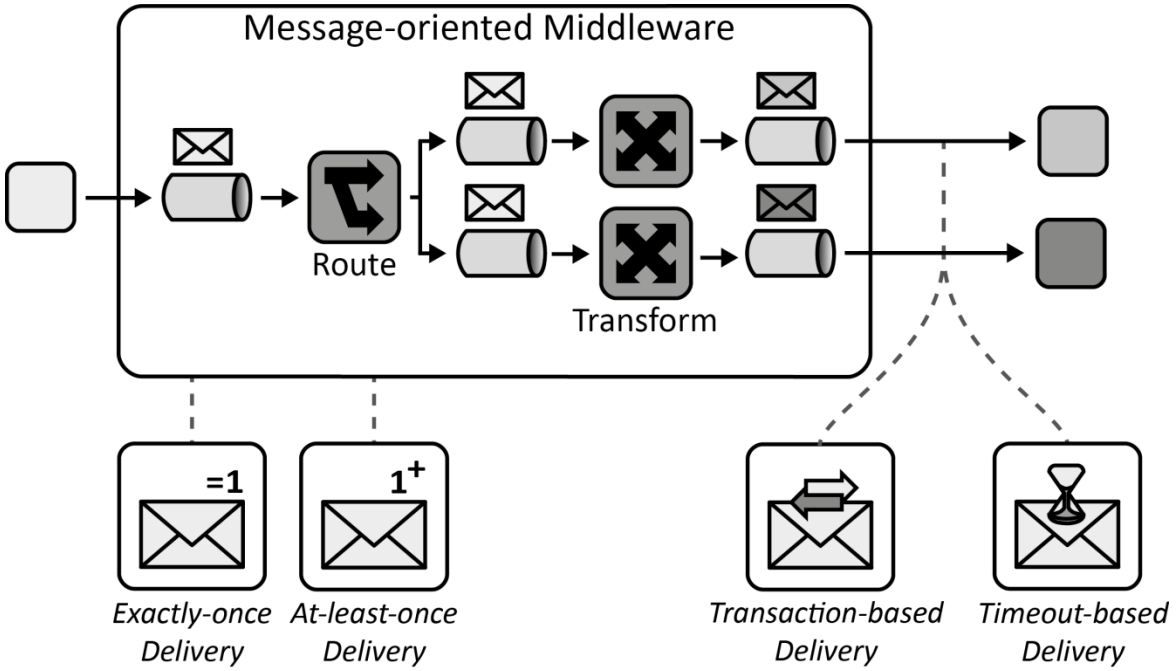






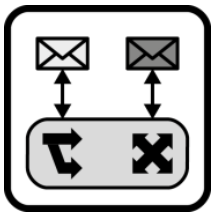






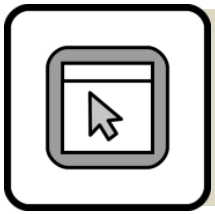
Message-oriented Middleware



Assumptions of communication partners are reduced (***Loose Coupling***)

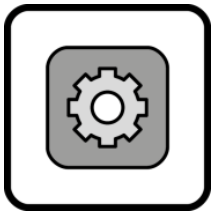
- **Platform**: implementation language used
- **Reference**: location of the communication partner (routing)
- **Time**: communication partners are active at different time / speed
- **Format**: message formats can change (transformation)

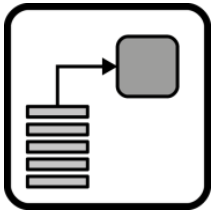


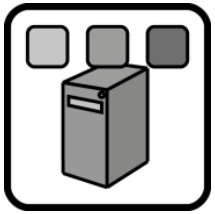


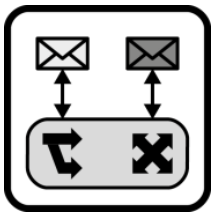
User Interface Component

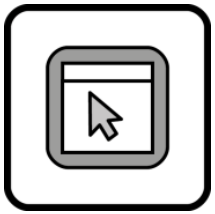
Synchronous user interfaces are accessed by humans, while application-internal interaction is realized asynchronously to ensure loose coupling.

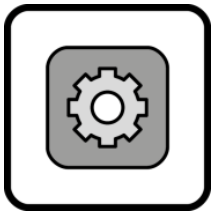


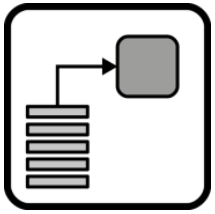


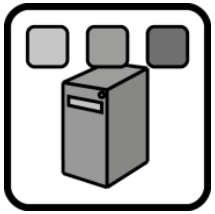




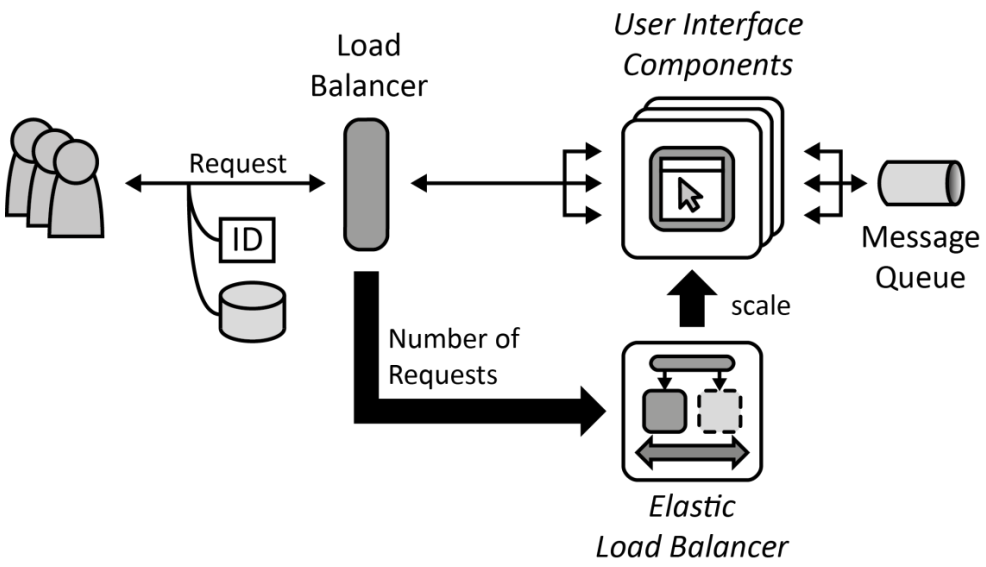


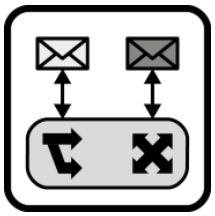


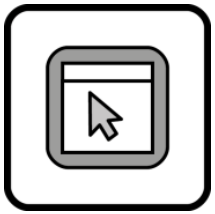


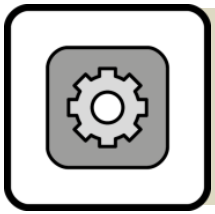


User Interface Component



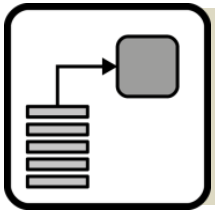






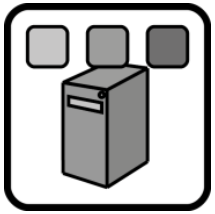
Processing Component

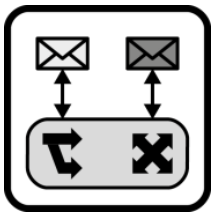
Processing functionality is handled by elastically scaled components.

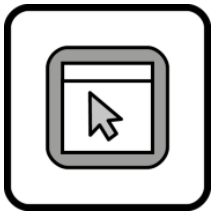


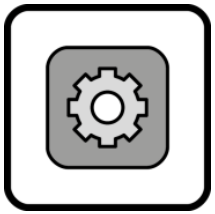
Batch Processing Component

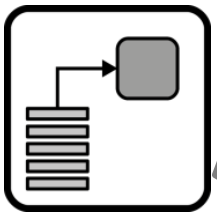
Requests are delayed until environmental conditions make their processing feasible.

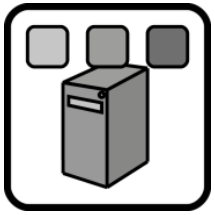




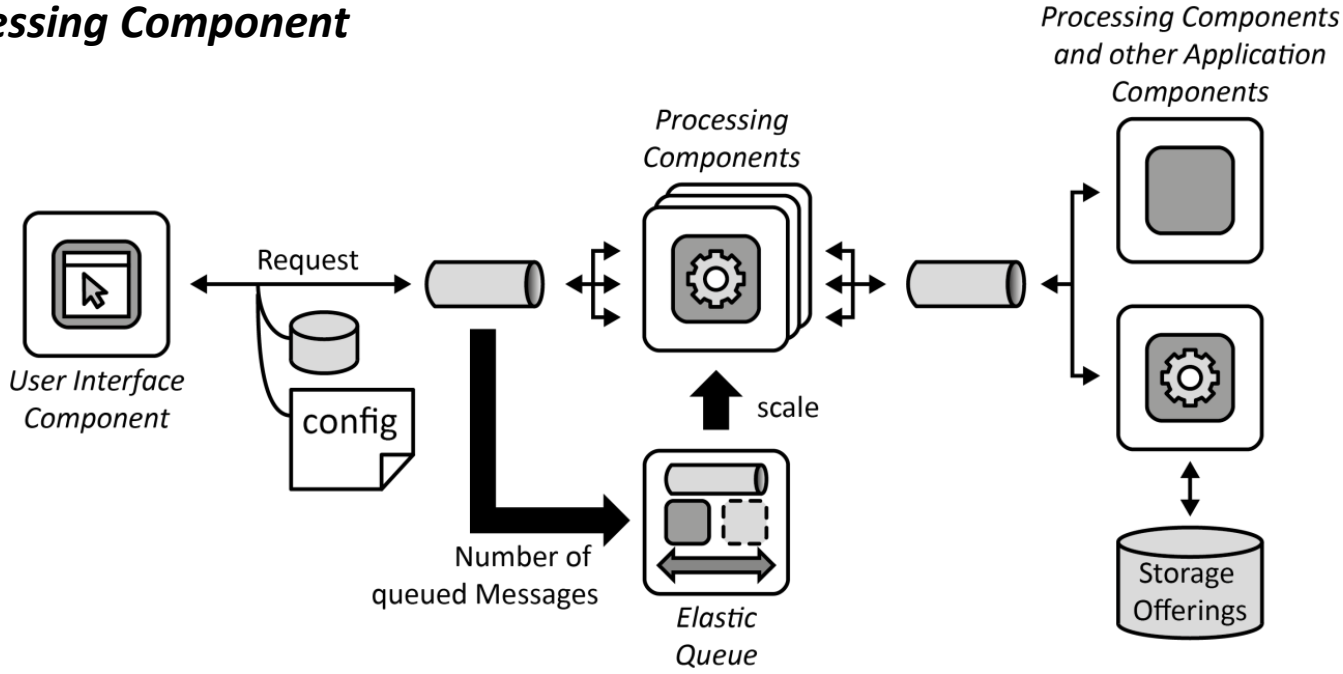




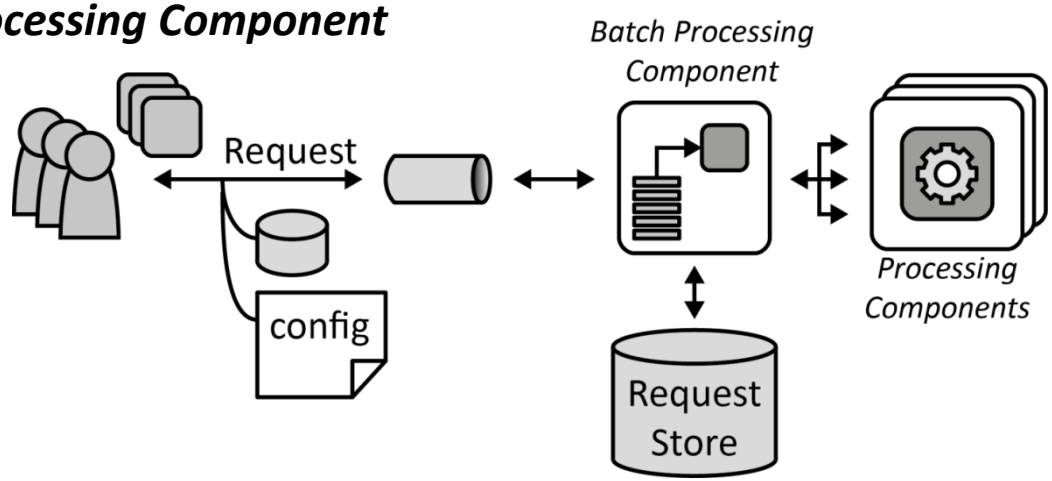


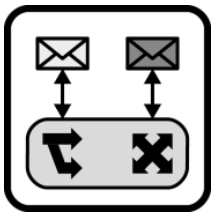


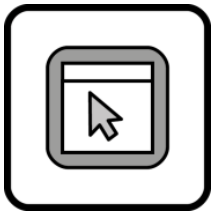
Processing Component

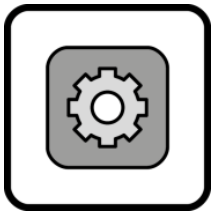


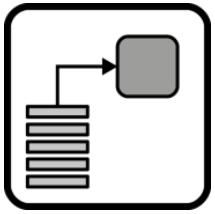
Batch Processing Component

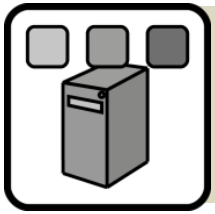






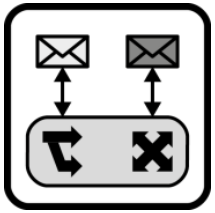


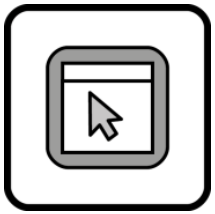


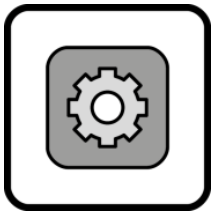


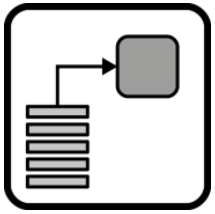
Multi-component Image

Virtual servers host multiple application components that may not be active at all times to reduce provisioning and decommissioning operations.

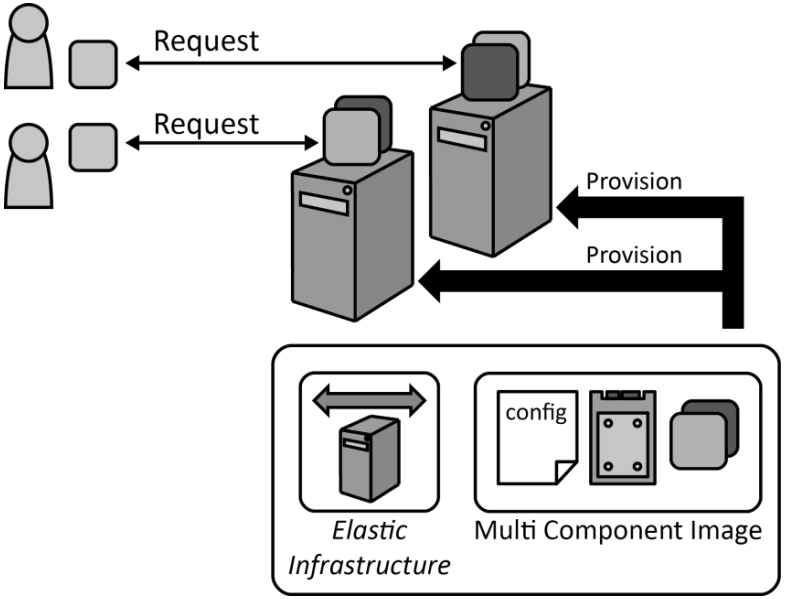






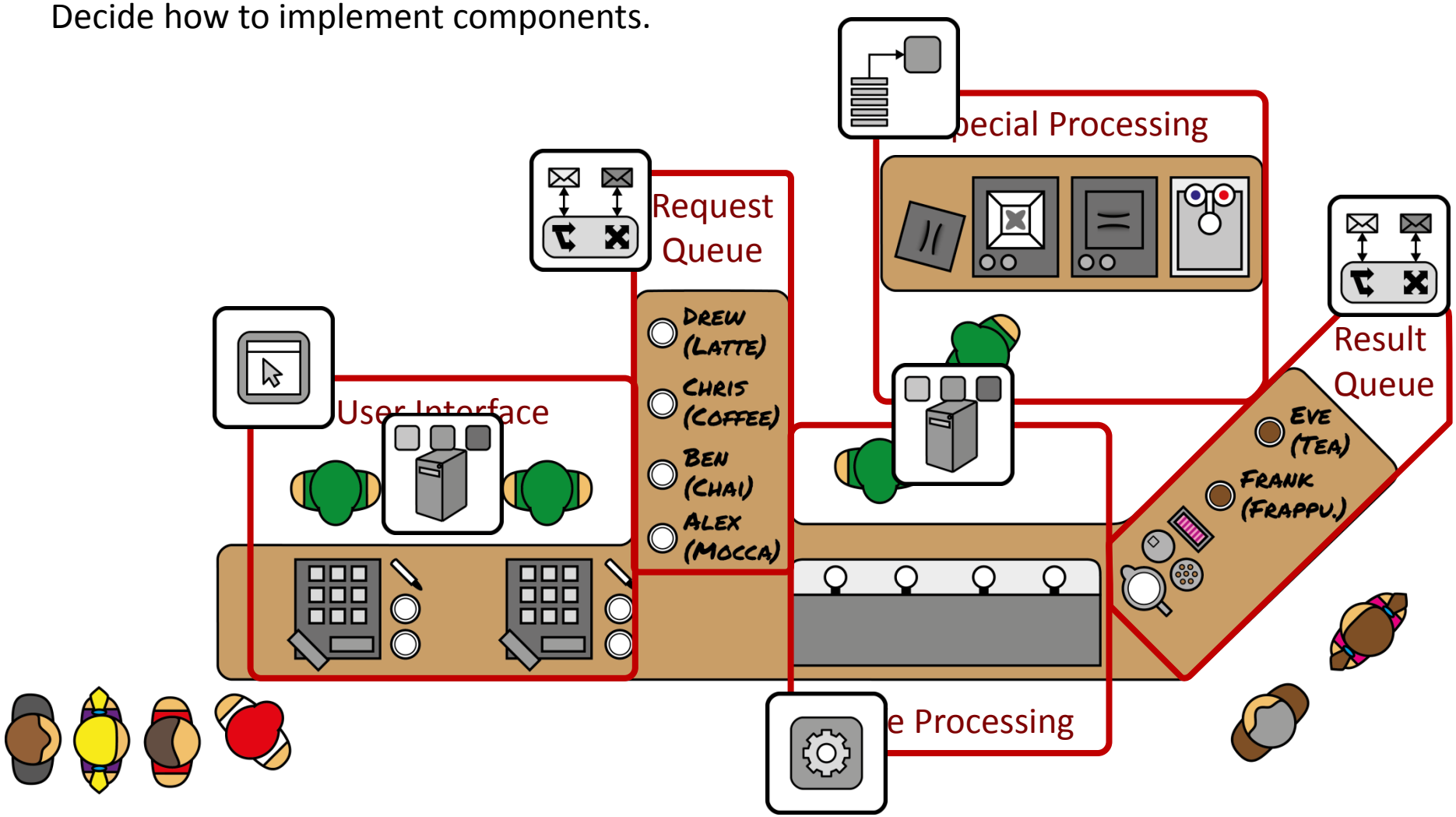


Multi-component Image



Coffee Shop – Refinement of Components

Decide how to implement components.

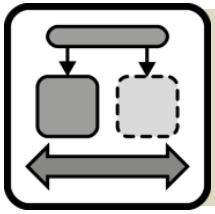


Lessons Learned

Components should be interact via messaging to ensure ***loose coupling***.

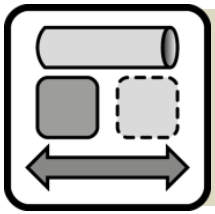
Resources supporting functional components should be flexible (*multi-component image*).

Elasticity and Resiliency



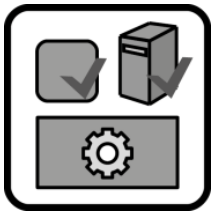
Elastic Load Balancer

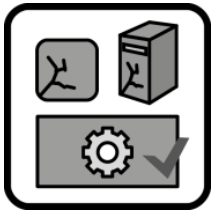
The number of synchronous accesses to an elastically scaled-out application is used to determine the number of required application component instances.

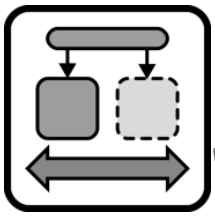
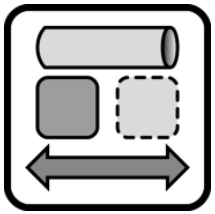
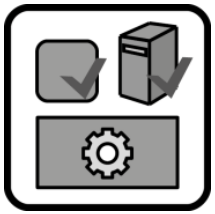
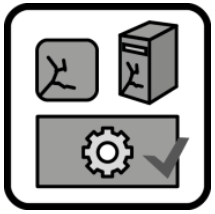


Elastic Queue

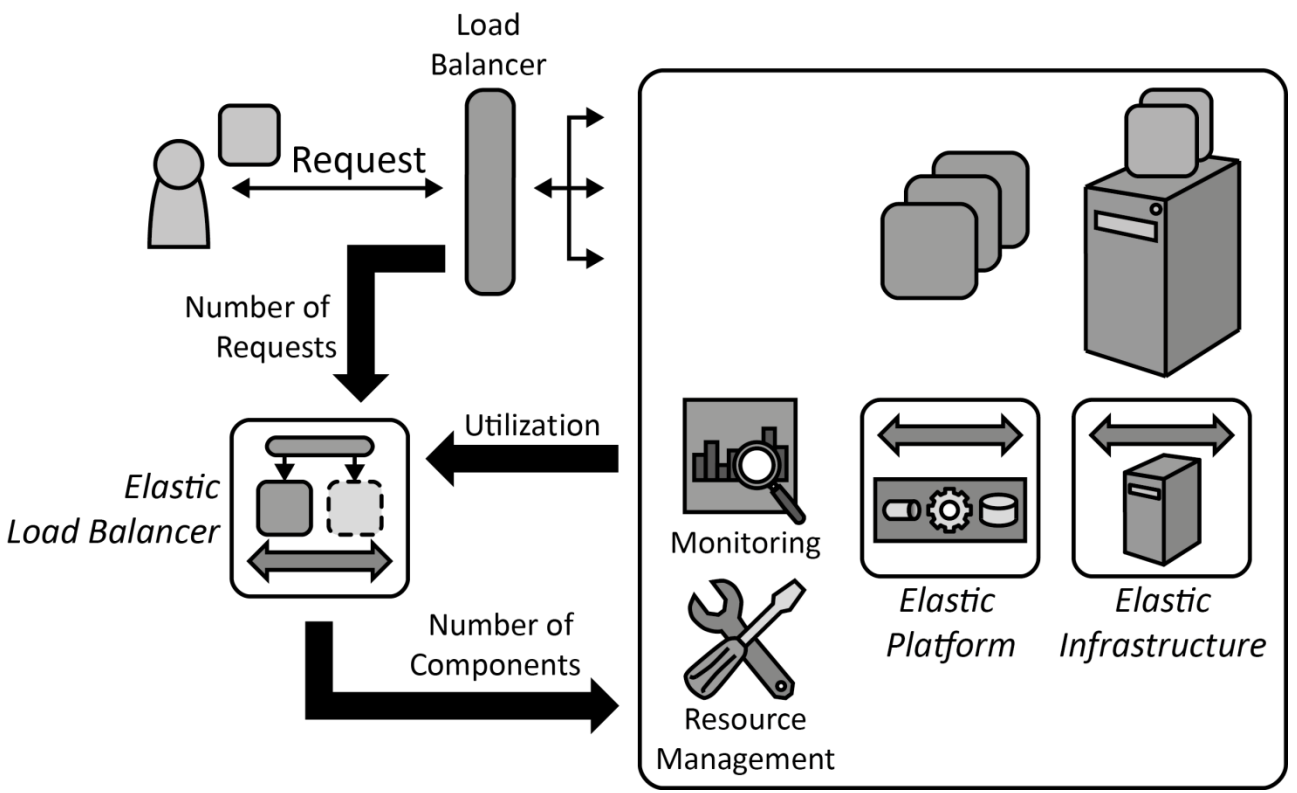
The number of accesses via messaging is used to adjust the number of required application component instances.

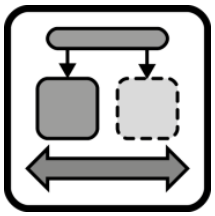
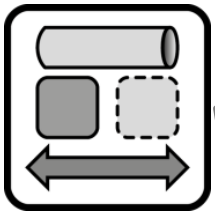
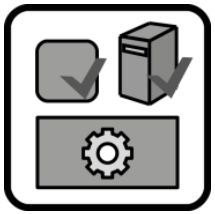
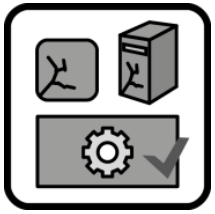


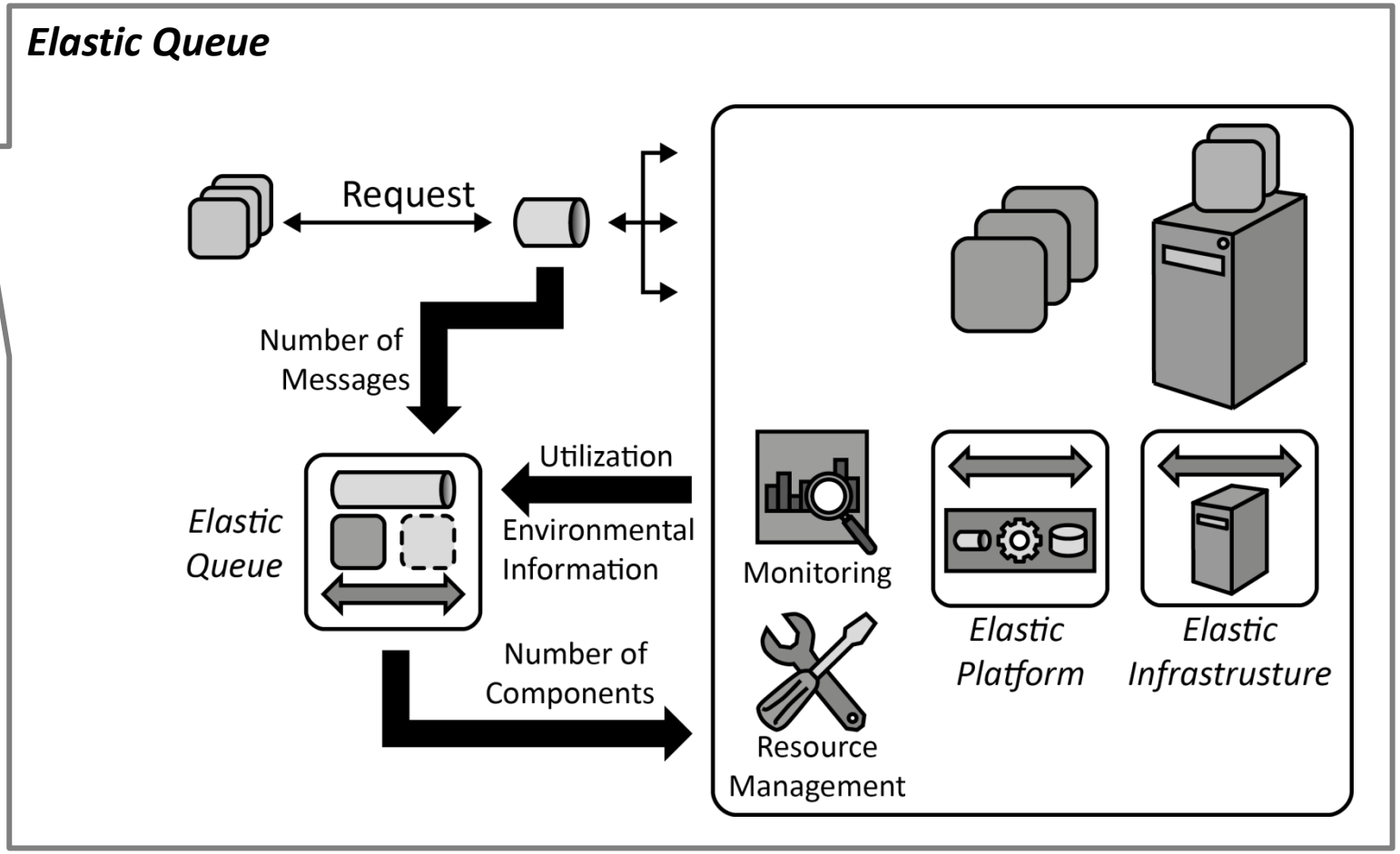


Elastic Load Balancer




Elastic Queue
Request
Number of Messages
Utilization
Environmental Information
Elastic Queue
Number of Components
Monitoring
Resource Management
Elastic Platform
Elastic Infrastructure

Pitfalls of Elasticity Management

Pitfalls of Elasticity Management

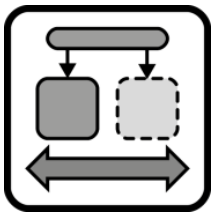
- **Configuration** of provider-supplied functionality commonly requires
  - **Condition**: monitored state to react upon.
  - **Time threshold**: duration how long the condition has been observed.
  - **Action**: tasks to be performed.

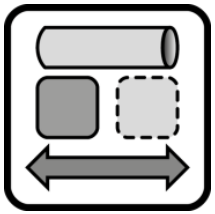
  (The same aspects have to be realized in custom implementation as well)
- These parameters are critical for the success of elasticity
  - **Bad conditions**: need to scale is missed / not seen.
  - **Bad threshold**: instances are added / removed too often.
  - **Inadequate actions**: too few instances are added / removed.
  - **Heuristics** have to be respected (holidays, seasons sales…)

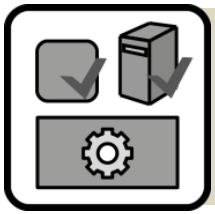
→ This behavior is very **similar to a heating system in buildings**

- Temperature is kept in a certain threshold
- Activations and deactivations of the heating shall be reduced!

Elasticity is hard to realize even if functionality is provider-supplied!

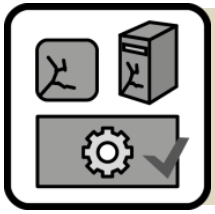






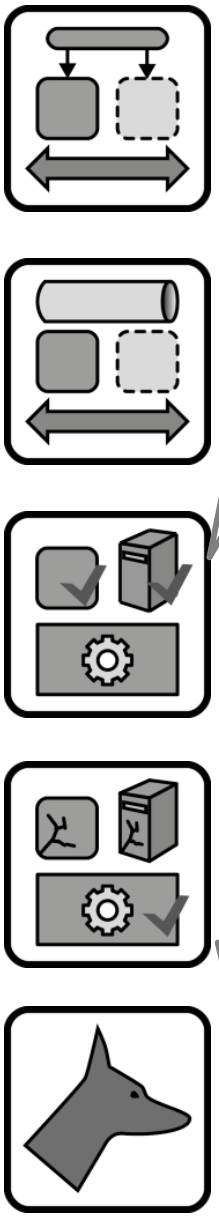
Node-based Availability

A cloud provider guarantees the availability of nodes, such as individual virtual servers, middleware components or hosted application components.

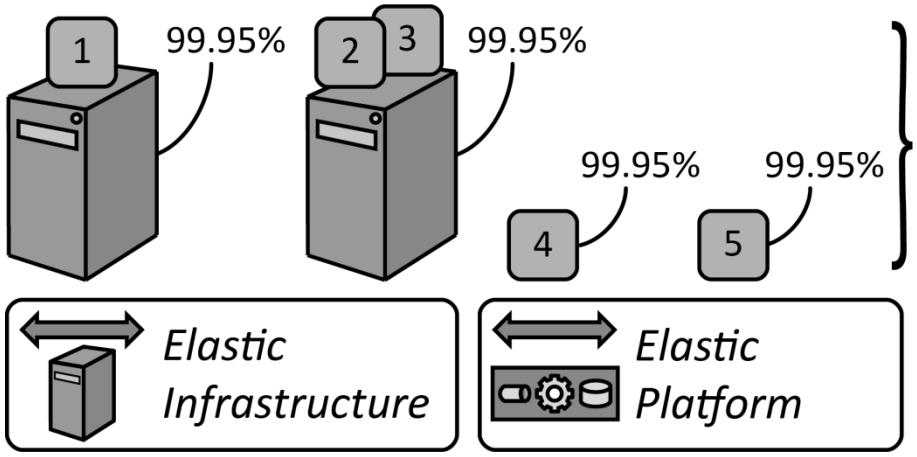


Environment-based Availability

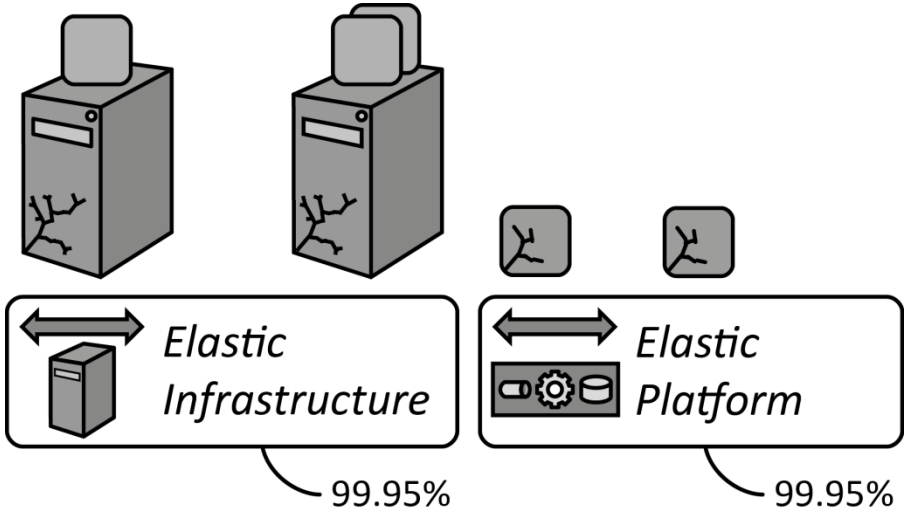
A cloud provider guarantees the availability of the environment hosting individual nodes, such as virtual servers or hosted application components.

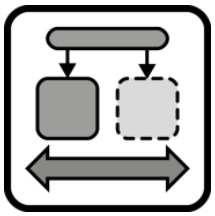


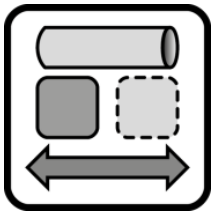
Node-based Availability

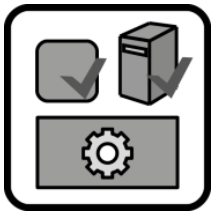


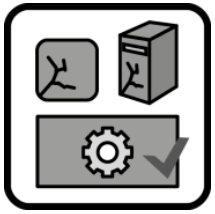
Environment-based Availability





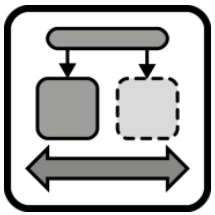


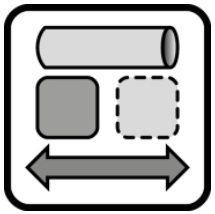


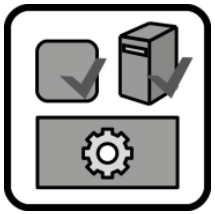


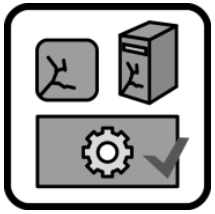
Watchdog

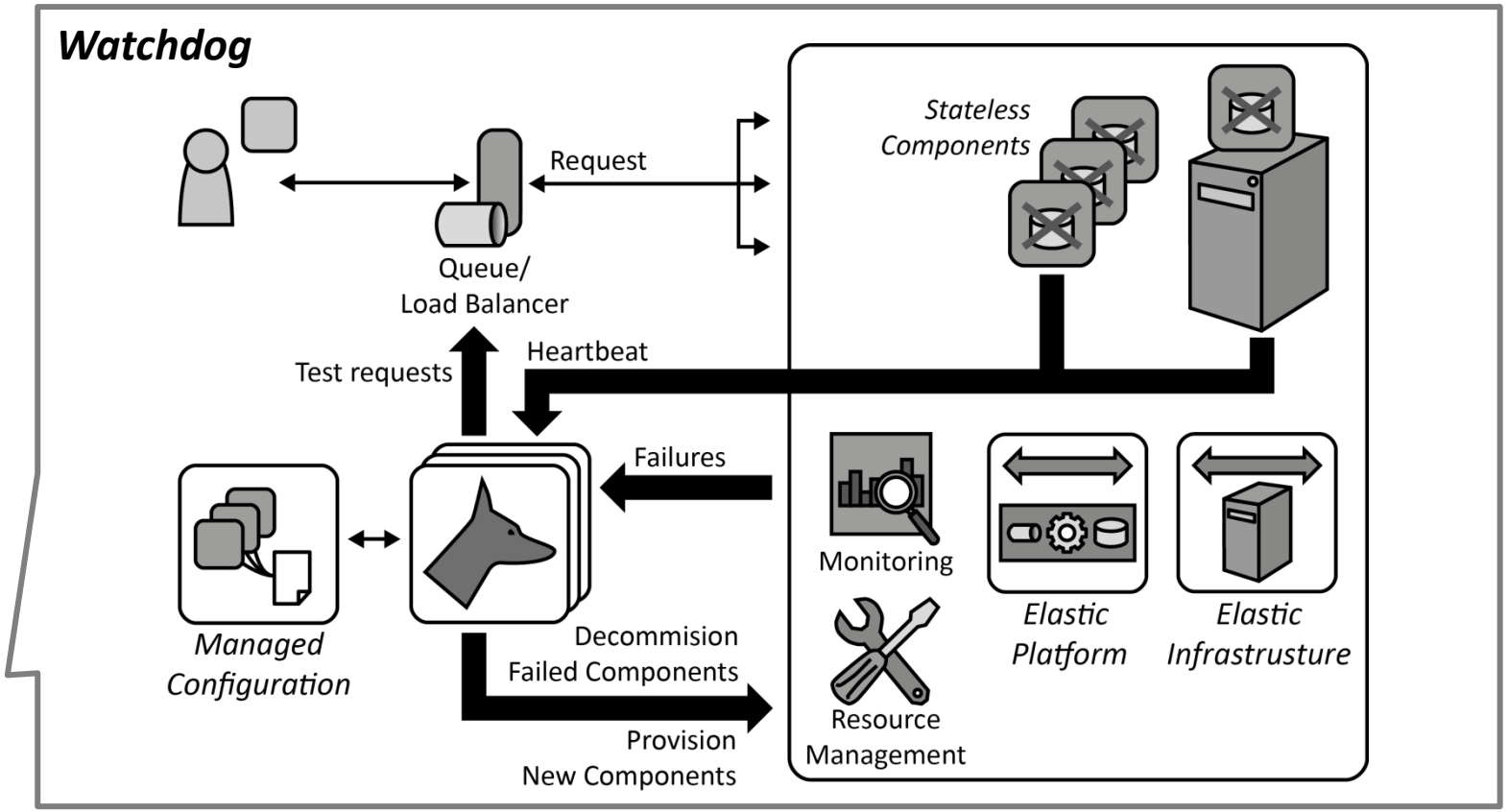
Applications cope with failures by monitoring and replacing application component instances if the provider-assured availability is insufficient.





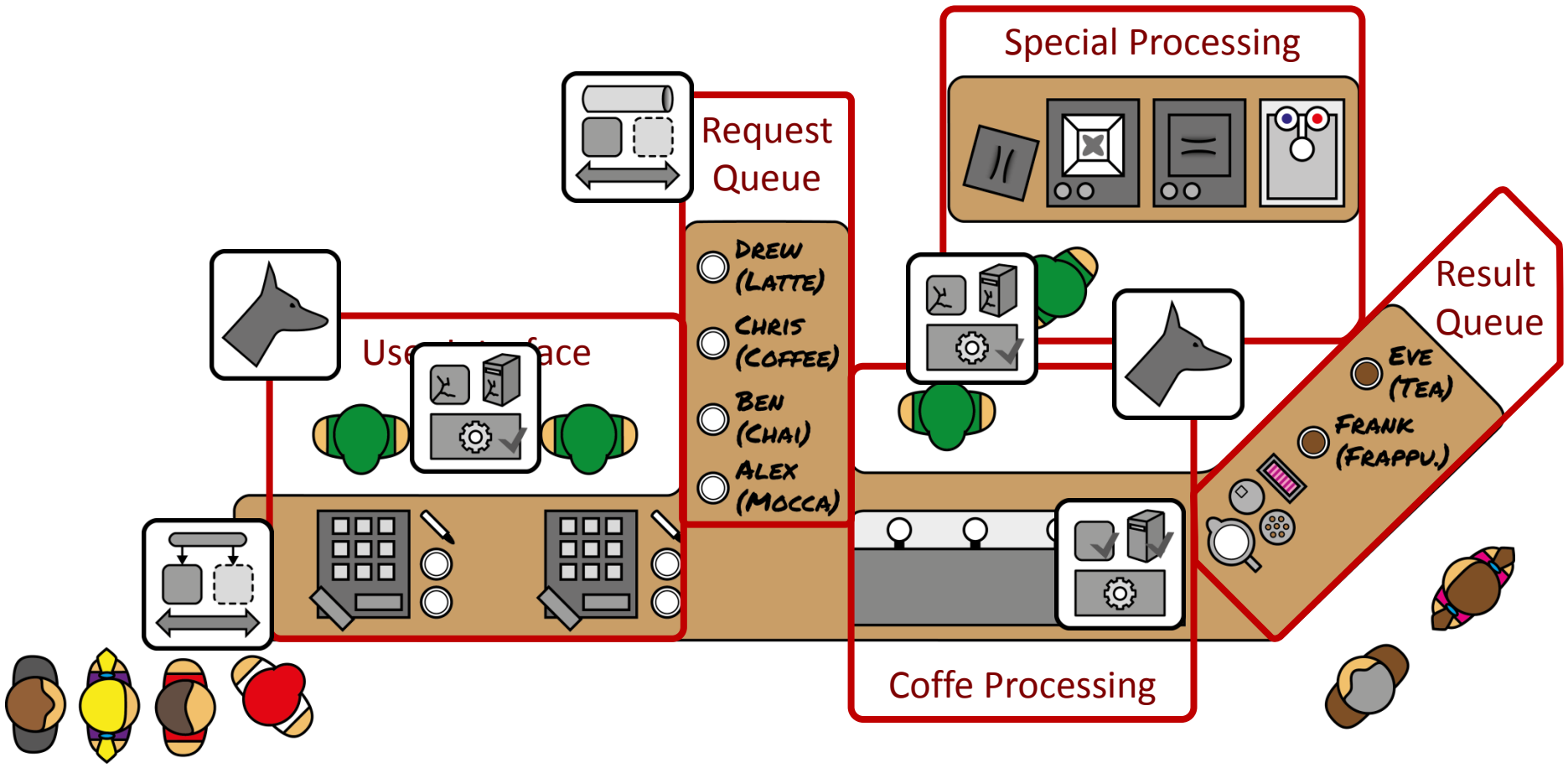





Watchdog
Request
Queue/
Load Balancer
Stateless
Components
Test requests
Heartbeat
Failures
Monitoring
Elastic
Platform
Elastic
Infrastrusture
Managed
Configuration
Decommision
Failed Components
Provision
New Components
Resource
Management

Elasticity and Resiliency

What shall happen if workload changes or something fails?



Lessons Learned

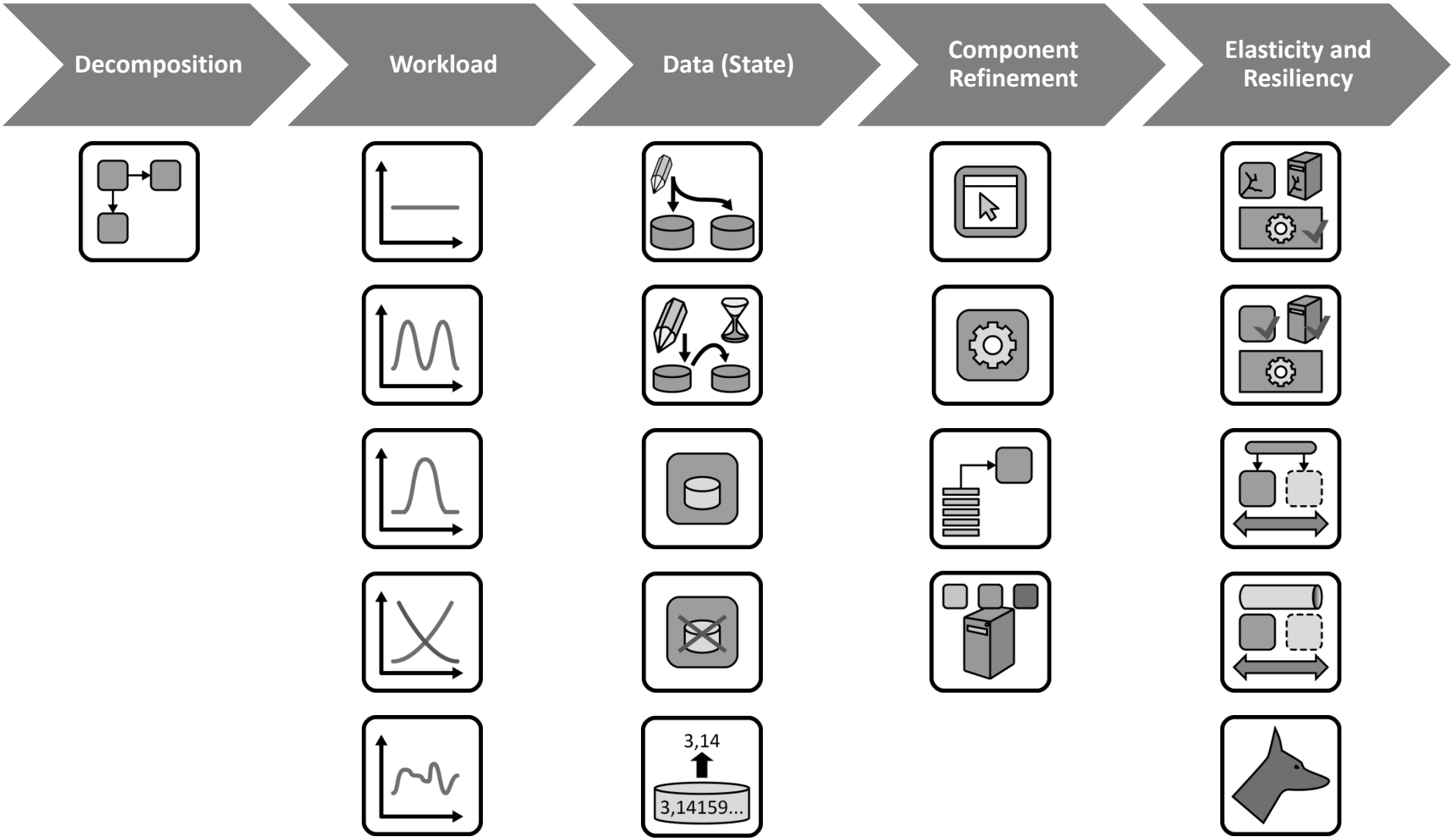
Analyze availability assured by provider
(*node-based availability* or *environment-based*).

In case of **low node-based availability** or **environment-based availability** implement a **watchdog**.

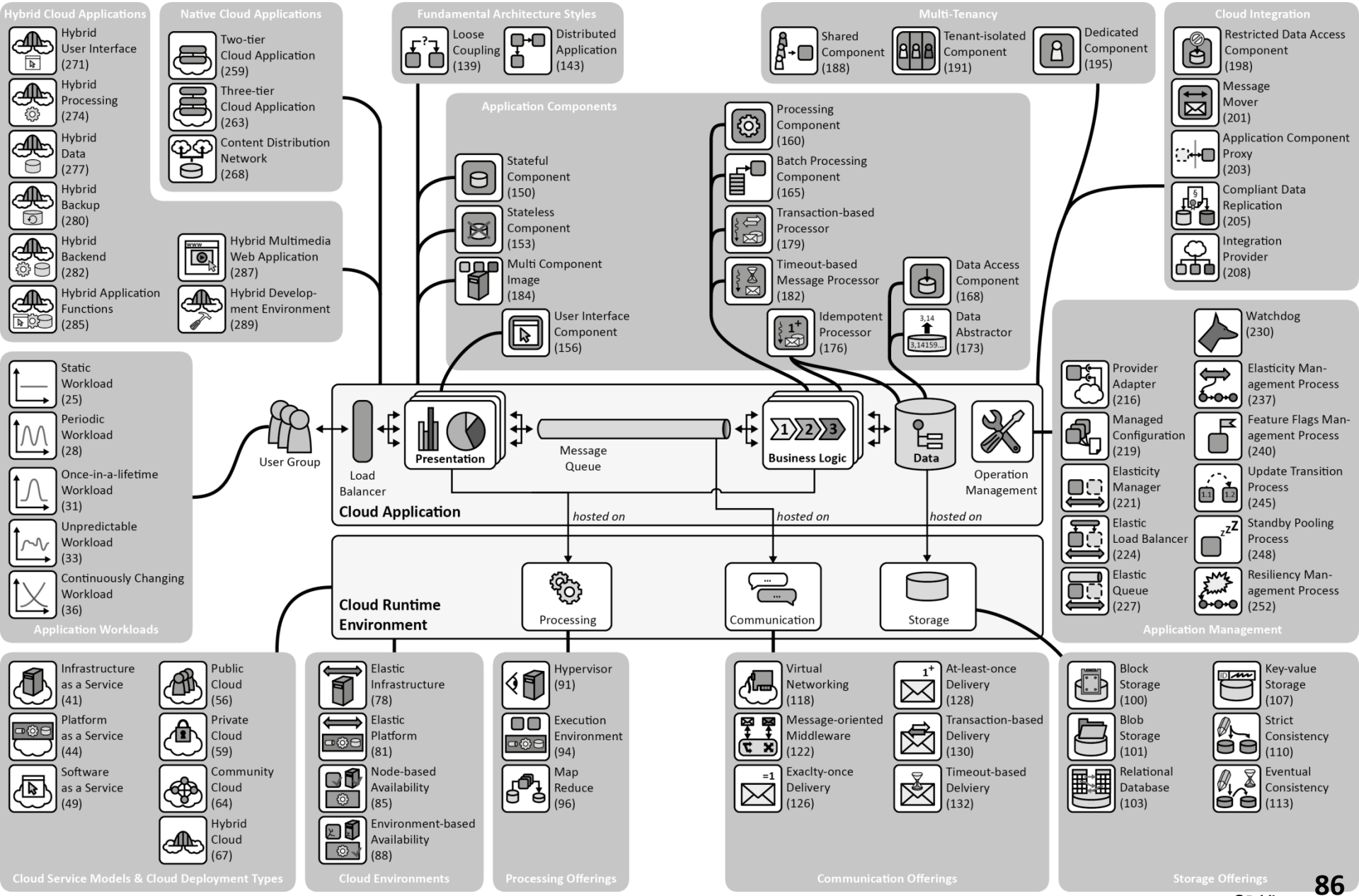
Use messages and requests to determine necessary component **instances**.

Summary

Summary Architectural Questions



And there are many more: **http://www.cloudcomputingpatterns.org**



http://www.springer.com/978-3-7091-1567-1