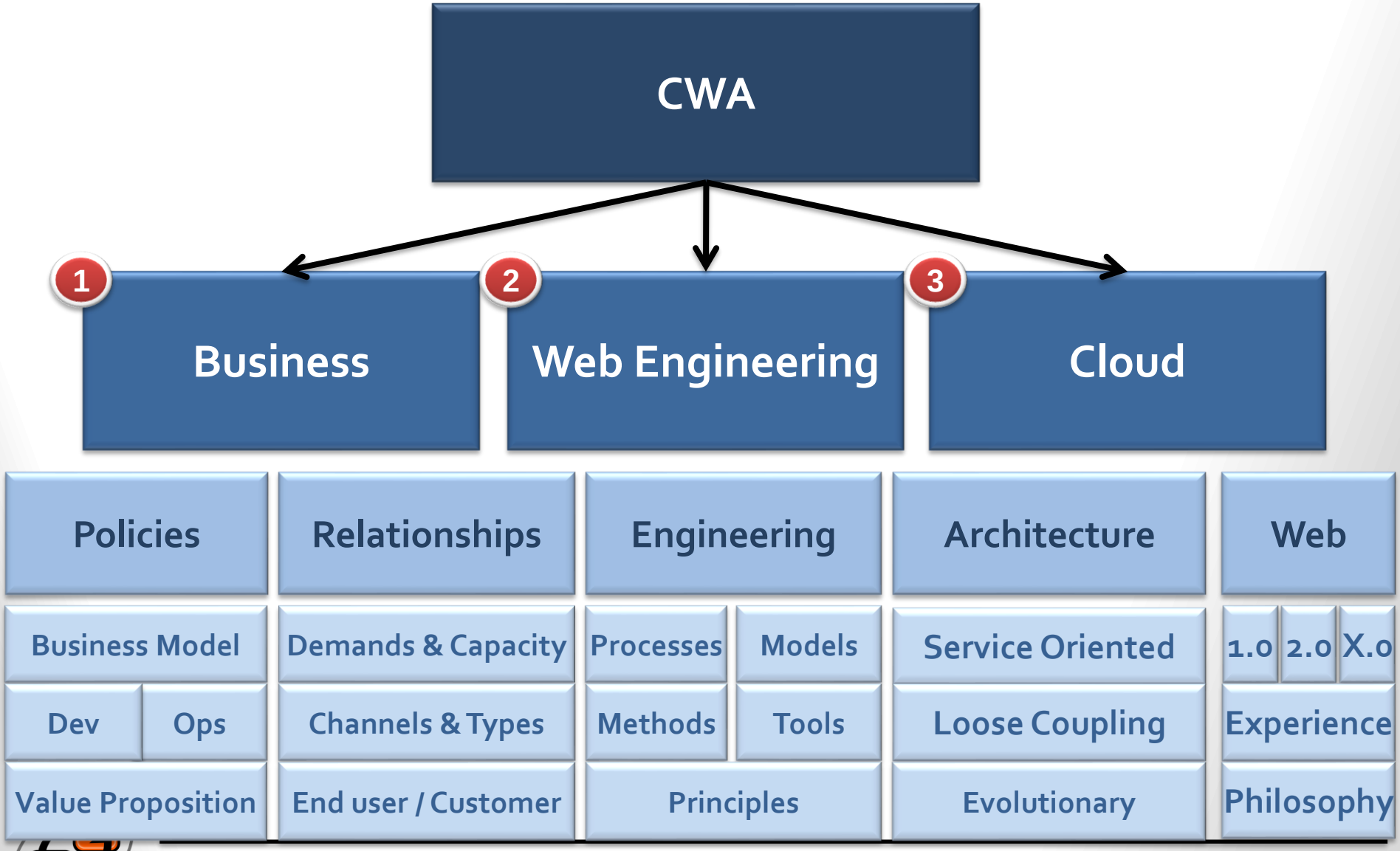
CORONA EMERGENCY LECTURE

Cloud & Web Anwendungen

Prof. Dr.-Ing. Martin Gaedke
Technische Universität Chemnitz
Fakultät für Informatik
Verteilte und selbstorganisierende Rechnersysteme

Lecture Outline



PART III

- Cloud Engineering

CHAPTER://19

- Cloud & Cloud Computing

The Cloud and Cloud Computing

- **What is the Cloud and Cloud Computing?**
  - It is more than technology – it is an *information technology paradigm* – *i*n other words: it is a distinct set of technologies, concepts, thought patterns, and standards that change the way we use, invest in and deploy technology solutions
  - It is based on sharing resources, providing services, and taking economies of scale into account

The Cloud and Cloud Computing

- **Why is it important?**
  - Instead of investing upfront in hardware, disk space, compute power or other information technology – users of cloud computing services can provision exactly the right type and size of computing resources they need at any given time
  - They can access as many resources as they need, almost instantly, and only pay for what you use – the resources *behave* in an **elastic** way

The Cloud and Cloud Computing

- **How is it done?**
  - By Virtualization: Requires a dedicated technology stack on top of (usually lots of!!) hardware for virtualization of resources (cpu, memory, disk space, network etc.)
  - Requires a business, operations, and sharing model for providing the resources on demand and as required
- Example
  - "Sharing a computer" – requires a host computer with a hypervisor (aka virtual machine monitor), which runs virtual machines using dedicated virtualization software, firmware and/or hardware, e.g., Xen, Hyper-V, VirtualBox. *Demo*: Parallels on Mac Computer running Windows
  - Virtualization of disk space and other infrastructure is possible in similar ways

Cloud Computing & Europe

Brussels, Jan 2010

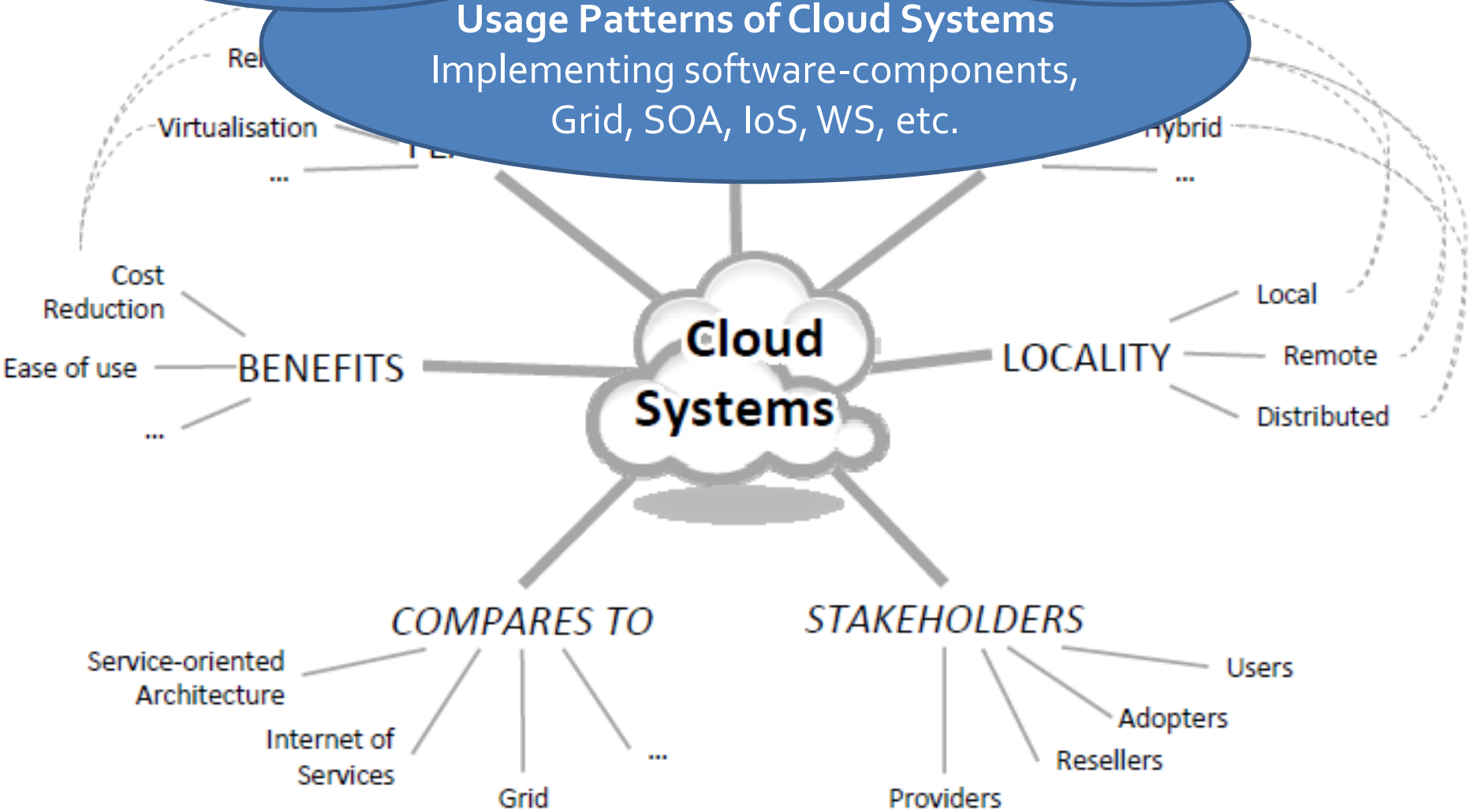
https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=1168
https://ec.europa.eu/newsroom//document.cfm?doc_id=1174

IaaS: Amazon S3 and EC2, SQL Azure…

PaaS: Google App Engine, Microsoft Azure…

SaaS: Google Docs, Salesforce, SAP…

Usage Patterns of Cloud Systems
Implementing software-components, Grid, SOA, IoS, WS, etc.



Quelle: http://cordis.europa.eu/fp7/ict/ssai/events-20100126-cloud-computing_en.html
See also: https://en.wikipedia.org/wiki/Cloud_computing

CHAPTER://20

- Cloud Technolgy and Economy

Usage, Sharing and Scale

Economic View

- ***Cost reduction***
- ***CAPEX vs. OPEX: costs for infrastructure transformed into operational costs***
- ***Time to Market***
- ***Pay per Use***
- ***Return on Investment (ROI)***
- ***Green-IT***
- ***Agile organization***
- etc

Technical View

- Elastic system
- Agility
- Technical flexibility
- **Availability**
- Reuseability
- Knowledge about usage
- Similar Services from different providers (can easily compete due to different payment models)

Quelle: http://cordis.europa.eu/fp7/ict/ssai/events-20100126-cloud-computing_en.html

What Is Availability?

- How much downtime can my organization afford without loosing productivity, profits, sales, etc.?
- The solution to High Availability is a combination of people, process, AND technology
  - Beware of 99.99% myth - The nines model does not take timing into account

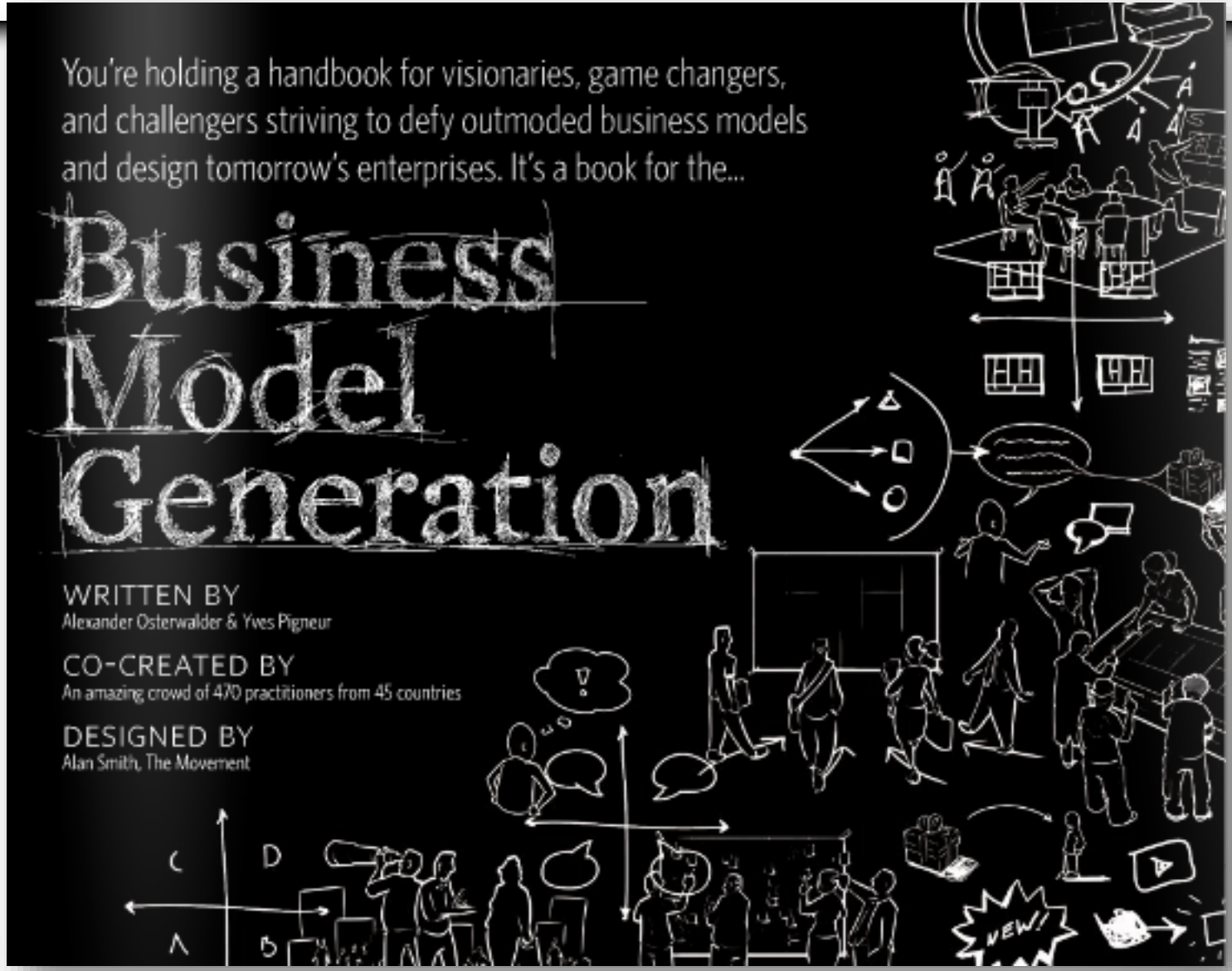
Availability:

A = (MTBF / (MTBF + MTTR)) * 100

- Mean Time Between Failure (**MTBF**) – average time a system is actually operational **[hours / failure count]**
- Mean Time To Recovery (**MTTR**) – average time needed to repair and restore service after failure **[Repair hours / failure count]**

- Simple Example:
  - 24/7 Web Site with two failure a week and each requires 1 hour
- On a year's time:

$$\frac{(52*7*24 / 52*2)}{(52*7*24 / 2) + 1/2} * 100 = 99.41\%$$


You're holding a handbook for visionaries, game changers, and challengers striving to defy outmoded business models and design tomorrow's enterprises. It's a book for the...
Business Model Generation
WRITTEN BY
Alexander Osterwalder & Yves Pigneur
CO-CREATED BY
An amazing crowd of 470 practitioners from 45 countries
DESIGNED BY
Alan Smith, The Movement
NEW!

The Business Model Canvas

Designed for:

Designed by:

On: Day Month Year

Iteration: No.

How?	What? (VP)	For whom?
Costs?		Revenues?

www.businessmodelgeneration.com

The Business Model Canvas

Designed for:

Designed by:

On: Day Month Year

Iteration: No.

How?

What? (VP)

Customer Relationships

What type of relationship does each of our Customer Segments expect us to establish and maintain with them?
Which ones have we established?
How are they integrated with the rest of our business model?
How costly are they?

EXAMPLES
Personal assistance
Dedicated Personal Assistance
Self-Service
Automated Services
Communities
Co-creation

Channels

Through which Channels do our Customer Segments want to be reached?
How are we reaching them now?
How are our Channels integrated?
Which ones work best?
Which ones are most cost-efficient?
How are we integrating them with customer routines?

CHANNEL PHASES:
1. Awareness
How do we raise awareness about our company's products and services?
2. Evaluation
How do we help customers evaluate our organization's Value Proposition?
3. Purchase
How do we allow customers to purchase specific products and services?
4. Delivery
How do we deliver a Value Proposition to customers?
5. After sales
How do we provide post-purchase customer support?

Customer Segments

For whom are we creating value?
Who are our most important customers?

Mass Market
Niche Market
Segmented
Diversified
Multi-sided Platfor

IaaS Customers

Costs?

Revenues?

www.businessmodelgeneration.com

The Business Model Canvas

Designed for:

Designed by:

On: Day Month Year

Iteration: No.

How?

Value Propositions

What value do we deliver to the customer?
Which one of our customer's problems are we helping to solve?

IaaS System

Customer Relationships

What type of relationship does each of our Customer Segments expect us to establish and maintain with them?
Which ones have we established?
How are they integrated with the rest of our business model?
How costly are they?

EXAMPLES
Personal assistance
Dedicated Personal Assistance
Self-Service
Automated Services
Communities
Co-creation

Customer Segments

For whom are we creating value?
Who are our most important customers?

Mass Market
Niche Market
Segmented
Diversified
Multi-sided Platfo

IaaS Customers

Channels

Through which Channels do our Customer Segments want to be reached?
How are we reaching them now?
How are our Channels integrated?
Which ones work best?
Which ones are most cost-efficient?
How are we integrating them with customer routines?

CHANNEL PHASES:
1. *Awareness*
How do we raise awareness about our company's products and services?
2. *Evaluation*
How do we help customers evaluate our organization's Value Proposition?
3. *Purchase*
How do we allow customers to purchase specific products and services?
4. *Delivery*
How do we deliver a Value Proposition to customers?
5. *After sales*
How do we provide post-purchase customer support?

Costs?

Revenues?

www.businessmodelgeneration.com

The Business Model Canvas

Designed for:

Designed by:

On: Day Month Year

Iteration: No.

How?

Value Propositions

What value do we deliver to the customer?
Which one of our customer's problems are we helping to solve?

IaaS System

Customer Relationships

What type of relationship does each of our Customer Segments expect us to establish and maintain with them?
Which ones have we established?
How are they integrated with the rest of our business model?
How costly are they?

EXAMPLES
Personal assistance
Dedicated Personal Assistance
Self-Service
Automated Services
Communities
Co-creation

Cha

Through wh
want to be
How are we
How are our Channels integrated?
Which ones work best?
Which ones are most cost-efficient?
How are w

CHANNEL PHASE
1. *Awareness*
How do we raise a
2. *Evaluation*
How do we help c
3. *Purchase*
How do we allow c
4. *Delivery*
How do we deliv
5. *After sales*
How do we provid

Web

Apps

Customer Segments

For whom are we creating value?
Who are our most important customers?

Mass Market
Niche Market
Segmented
Diversified
Multi-sided Platfo

IaaS Customers

Costs?

Revenues?

www.businessmodelgeneration.com

The Business Model Canvas

Designed for:

Designed by:

On: Day Month Year

Iteration: No.

Key Partners

Who are our Key Partners?
Who are our key suppliers?

- Energy Provider
- Hardware Provider
- Network Provider
- Admins
- Developers

Key Activities

- IaaS

Key Resources

What Key Resources do our Value Propositions require?
Our Distribution Channels? Customer Relationships?
Revenue Streams?

TYPES OF RESOURCES
Physical
Intellectual (brand patents, copyrights, data)
Human
Financial

- Data Center

Value Propositions

What value do we deliver to the customer?
Which one of our customer's problems are we helping to solve?

- IaaS System

Customer Relationships

What type of relationship does each of our Customer Segments expect us to establish and maintain with them?
Which ones have we established?
How are they integrated with the rest of our business model?
How costly are they?

EXAMPLES
Personal assistance
Dedicated Personal Assistance
Self-Service
Automated Services
Communities
Co-creation

Cha

Through wh
want to be
How are we
How are our Channels integrated?
Which ones work best?
Which ones are most cost-efficient?
How are w

CHANNEL PHASE
1. *Awareness*
How do we raise a
2. *Evaluation*
How do we help c
3. *Purchase*
How do we allow c
4. *Delivery*
How do we deliv
5. *After sales*
How do we provid

- Web
- Apps

Customer Segments

For whom are we creating value?
Who are our most important customers?

Mass Market
Niche Market
Segmented
Diversified
Multi-sided Platfo

- IaaS Customers

Costs?

Revenues?

www.businessmodelgeneration.com

The Business Model Canvas

Designed for:

Designed by:

On: Day Month Year

Iteration: No.

Key Partners

Who are our Key Partners?
Who are our key suppliers?

- Energy Provider
- Hardware Provider
- Network Provider
- Admins
- Developers

Key Activities

- IaaS

Key Resources

What Key Resources do our Value Propositions require?
Our Distribution Channels? Customer Relationships?
Revenue Streams?

TYPES OF RESOURCES
Physical
Intellectual (brand patents, copyrights, data)
Human
Financial

- Data Center

Value Propositions

What value do we deliver to the customer?
Which one of our customer's problems are we helping to solve?

- IaaS System

Customer Relationships

What type of relationship does each of our Customer Segments expect us to establish and maintain with them?
Which ones have we established?
How are they integrated with the rest of our business model?
How costly are they?

EXAMPLES
Personal assistance
Dedicated Personal Assistance
Self-Service
Automated Services
Communities
Co-creation

Cha

Through wh
want to be
How are we
How are our Channels integrated?
Which ones work best?
Which ones are most cost-efficient?
How are w

CHANNEL PHASE
1. *Awareness*
How do we raise a
2. *Evaluation*
How do we help c
3. *Purchase*
How do we allow c
4. *Delivery*
How do we deliver
5. *After sales*
How do we provid

- Web
- Apps

Customer Segments

For whom are we creating value?
Who are our most important customers?

Mass Market
Niche Market
Segmented
Diversified
Multi-sided Platfo

- IaaS Customers

Cost Structure

What are the most important costs inherent in our business model?
Which Key R
Which Key A

IS YOUR BUSINESS
Cost Driven (leanes
Value Driven (focus

SAMPLE CHARACTE
Fixed Costs (salaries
Variable costs
Economies of scale
Economies of scope

- Energy
- Hardware
- Staff costs

Revenues?

www.businessmodelgeneration.com

The Business Model Canvas

Designed for:

Designed by:

On: Day Month Year

Iteration: No.

Key Partners

Who are our Key Partners?
Who are our key suppliers?

- Energy Provider
- Hardware Provider
- Network Provider
- Admins
- Developers

Key Activities

- IaaS operations

Key Resources

What Key Resources do our Value Propositions require?
Our Distribution Channels? Customer Relationships?
Revenue Streams?

TYPES OF RESOURCES
Physical
Intellectual (brand patents, copyrights, data)
Human
Financial

- Data Center

Value Propositions

What value do we deliver to the customer?
Which one of our customer's problems are we helping to solve?

- IaaS

Customer Relationships

What type of relationship does each of our Customer Segments expect us to establish and maintain with them?
Which ones have we established?
How are they integrated with the rest of our business model?
How costly are they?

EXAMPLES
Personal assistance
Dedicated Personal Assistance
Self-Service
Automated Services
Communities
Co-creation

Cha

Through wh
want to be
How are we
How are our Channels integrated?
Which ones work best?
Which ones are most cost-efficient?
How are w

CHANNEL PHASES
1. *Awareness*
How do we raise a
2. *Evaluation*
How do we help c
3. *Purchase*
How do we allow c
4. *Delivery*
How do we deliver
5. *After sales*
How do we provid

- Web
- Apps

Customer Segments

For whom are we creating value?
Who are our most important customers?

Mass Market
Niche Market
Segmented
Diversified
Multi-sided Platfo

- IaaS Customers

Cost Structure

What are the most important costs inherent in our business model?
Which Key R
Which Key A

IN YOUR BUSINESS M
Cost Driven (leanes
Value Driven (focus

SAMPLE CHARACTER
Fixed Costs (salaries
Variable costs
Economies of scale
Economies of scope

- Energy
- Hardware
- Staff costs

Revenue Streams

For what value are our customers really willing to pay?
For what do they currently pay?
How are they currently paying?
How would they prefer to pay?
How much does each Revenue Stream contribute to over

TYPES:
Asset sale
Usage fee
Subscription Fees
Lending/Renting/Leasing
Licensing
Brokerage fees
Advertising

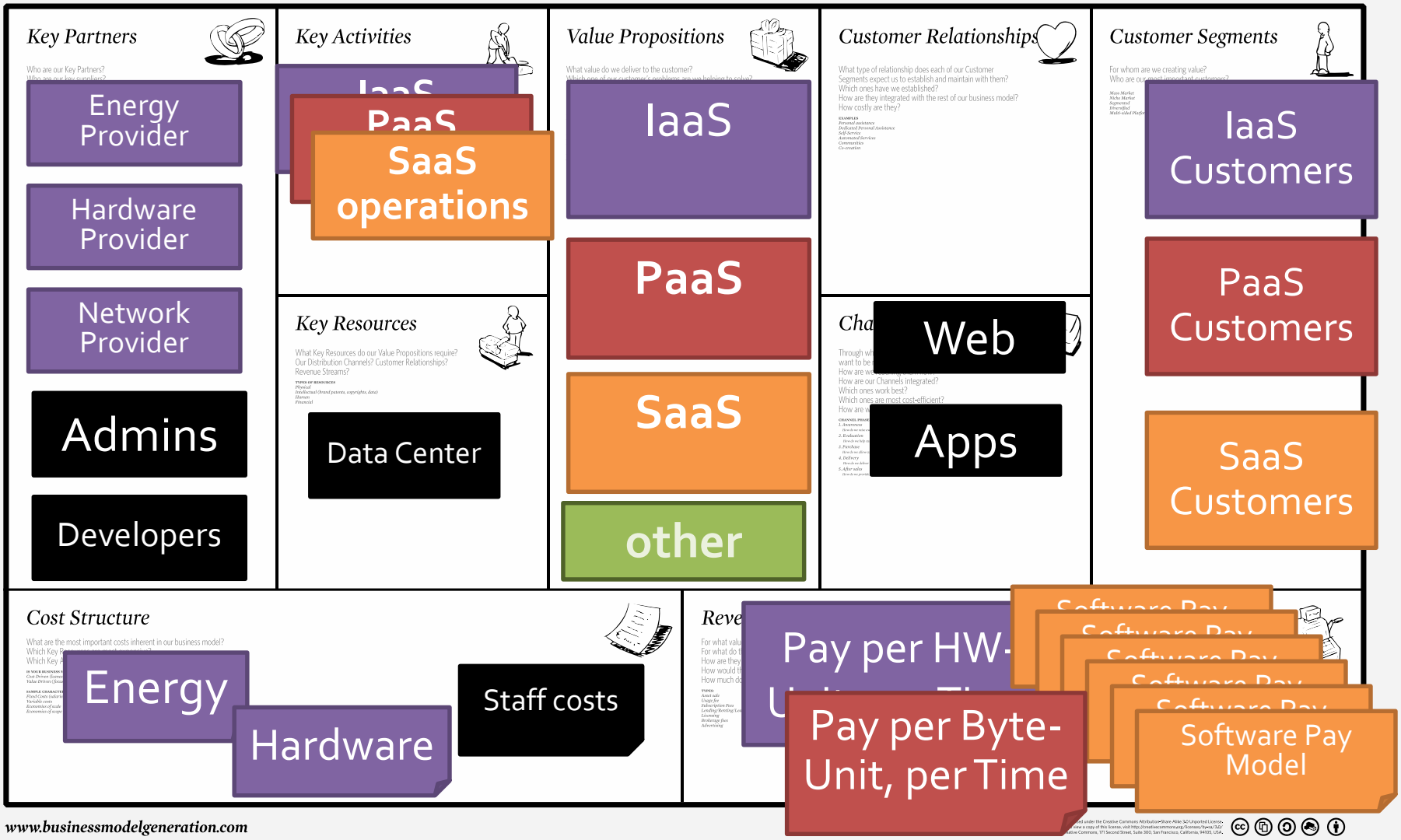
FIXED PRICING
List Price
Product feature dependent
Customer segment dependent
Volume dependent

DYNAM
Negoti
Yield M
Real-t

- Pay per HW-Unit, per Time

www.businessmodelgeneration.com

Economic Aspects and the Cloud



CHAPTER://21

- Build and Test – "Creating the solution"

Introduction

- Goal
  - Transform FuncSpec to real code
  - Develop final documentation
  - Implementation of the solution
- Challenges
  - Mapping from design to code
  - Development in parallel to deployment and operations
  - Tests

Non-Exhaustive Tec-Map

Systems Technologies

Data
- SQL
- File
- XML

Process
- Web Service
- Servlets
- Components
- XSLT

DSA
- HTTP, Cookie
- SMTP
- Web Service, UDDI
- SOAP, WS-*
- WSA

U I X Technologies

Dialogue
- HTML
- XHTML
- XForms
- Applets

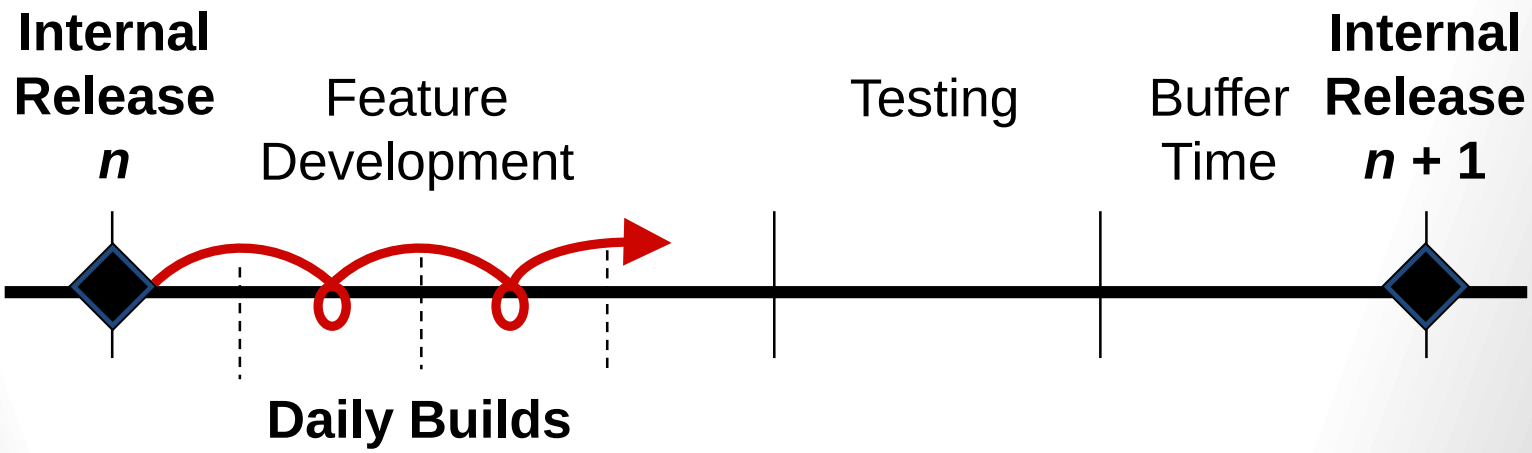
Presentation
- HTML, XHTML
- XSLT
- RSS
- Images, Audio
- Diverse Mime
- WAI-Guidelines

Navigation
- HTML
- XPath
- XPointer
- XLInk
- RDF

Further help: Cf. supporting standards and guidelines, like IEEE Std 2001-2002

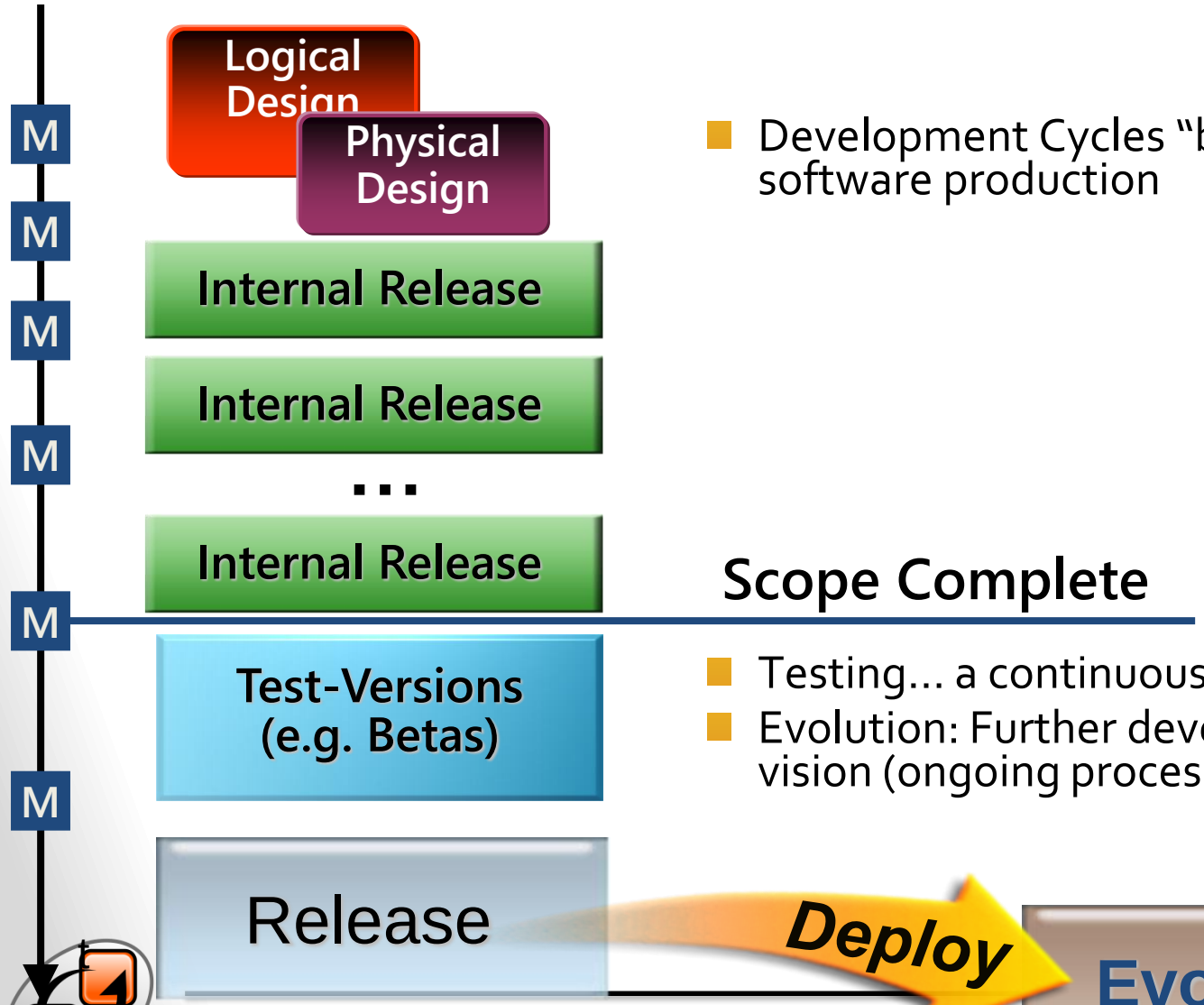
Building Releases…

- Daily builds or Continuous Integration
  - A way to make the product and its progress visible
  - The *heartbeat* of the development process
  - Components are the unit of deployment (aka container)



And a lot of other core tools and methods like in software development should be used – not to forget Version Control

Testing & Evolution in Context

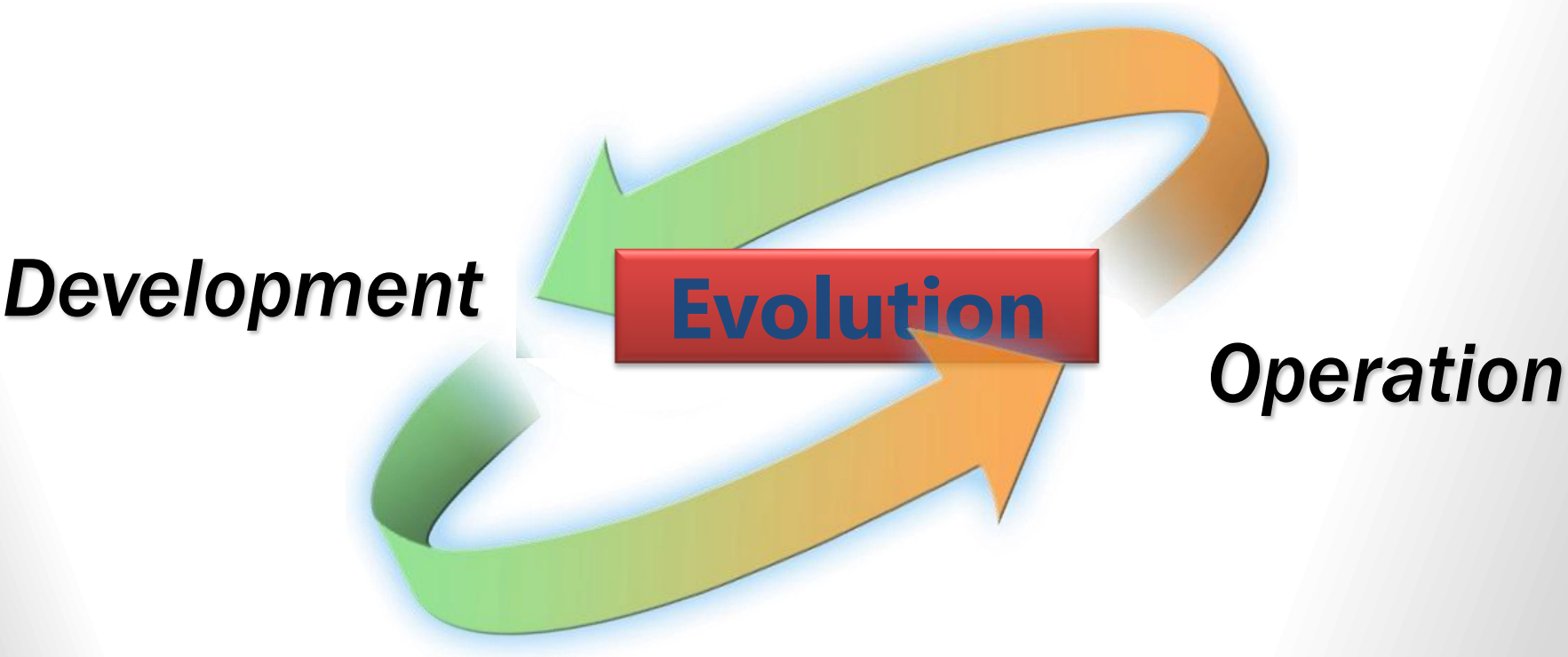


- Development Cycles "behave" like in common software production

- Testing… a continuous process…
- Evolution: Further developments towards the vision (ongoing process)

Evolution…

… continuous progress at DevOps level
and at Business and Development level



CHAPTER://22

- Testing and Monitoring

Introduction

- Testing is extremely difficult
  - Testing is a continuous process
  - Starts during Assessment Phase
  - Address Testing seriously! Prepare for Test Plans
- Early:
  - Requirements for testing
  - Criteria for non-functional requirements – How to test for "good"?
- Later:
  - Define Test Cases
  - Unit Tests, Acceptance Tests, UI Tests etc.
- Final:
  - Release Test Criteria (e.g. ZBB, Customer Feedback, etc.)
  - Based on Criteria an Internal Release becomes a Release

Testing Problems

- Things to look at – and define Test Plans for:
  - Spelling errors, broken links, buggy scripts
  - User and eCommerce night mares: overcharging accounts
  - Assumption of Correctness "site is correct because it looks and loads right"-Syndrome
  - **The scenario: App/Browser Types x PlugIns x Script Engines x OS x Hardware x Network Connections**
- BTW, Job of Testing – a social problem…
  - Testing is a difficult job and in many cases thankless
  - Proper testing is often not understood or appreciated and often seen as a boring task

Test Plans and Procedures

- Prepare Test Plans
  - Functionality Testing
  - Content Testing
  - User Testing
  - Security Testing
- If applicable try to use Test Labs

- Procedures for finding issues/problems
  - Report Problem Tracking System
  - **Track, Handle, Finalize**: Initiate Change Request
  - Be integral part of Configuration Management

Functionality Testing

- Site functions properly and meets specification
- Main Testing: Units, integration (all units together), browser
- Final Testing: User's system configuration (e.g. speed of hard-drives, Java runtime with different processor speeds), delivery (network and server aspects)
- **Test Labs** may help in some cases, especially for final testing

Content Testing

- Content of site is correctly implemented
- Consider proof-reading, especially spelling of names and companies
- Check for copyright inclusion and legal disclaimers
- Check images and other media type (includes consideration of user's system configuration)

User Testing

- Site meets user's needs and is usable
- If available testing rooms
- Low-cost testing with some people and questionnaires
- Online-testing with feedback option ("send us your comments and you may win…")

Security Testing

- Should not be part of functionality testing
- Handle explicitly
- Include application-, server-, network-, physical site-security, and physical access by Staff; as well as many other issues

Handling Results Of Testing

- Use a Problem Tracking System
  - **Track**: Problem (Id, problem description, discovered by, when, user's system configuration, severity,…)
  - **Handle**: ProblemId, HandledBy, Status, …
  - **Finalize**: Initiate Change Request (**CR**)
- Should be part of a change request processing approach, cf. Requirements Engineering

Bobby

WebEngineering.org Community - The Site about Web Engineering - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media

Address http://bobby.watchfire.com/bobby/bobbyServlet?URL=http%3A%2F%2Fwebengineering.org&output=Submit& Go

Google website" usability ibm.com Search Web Search Site Page Info Up Highlight

Bobby™

© Copyright 2002 Watchfire Corporation

Skip to report

URL tested: http://webengineering.org, February 12, 2003 4:37:25 PM EST
Watchfire Bobby Core v4.0.1, WAI Content Accessibility Guidelines 1999/05/05, Support Level: AAA

Note: To ensure that the marked up report is readable, all stylesheets and scripts are removed. The page may not appear as expected.

Internet

http://bobby.watchfire.com/bobby/html/en/index.jsp

Web Application Test Tools

ACTSamples - Microsoft Application Center Test

File Edit View Actions Help

ACTSamples
- Tests
  - Browser Type
  - Body search
  - Duwamish v7.0
  - Fitch & Mather v7.0
  - Form data
  - TestFiles
- Results
- Users
  - Default
  - Fitch & Mather Users Grou
  - Duwamish Users Group

Test runs:
- Browser Type
  - report-Browser Type (Example 4)
  - report-Browser Type (Example 3)
  - report-Browser Type (Example 2)

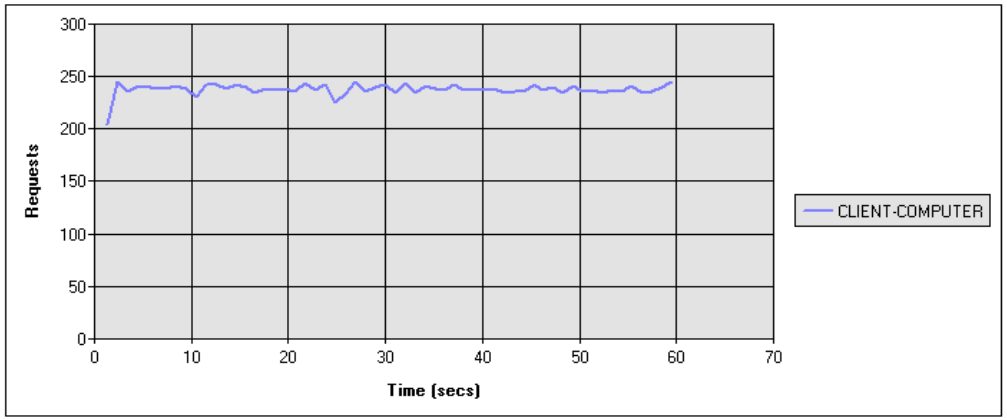
Report: Overview

Summary
Performance Counters

Application Center Test
Overview: Summary

Test Name:	ACTSamples: Browser Type
Test Run Name:	report-Browser Type (Example 3)
Test Started:	15.06.2001 02:35:00
Test Duration:	00:00:01:00
Test Iterations:	3.562
Test Notes:	-

Test Run Graph



Show Errors

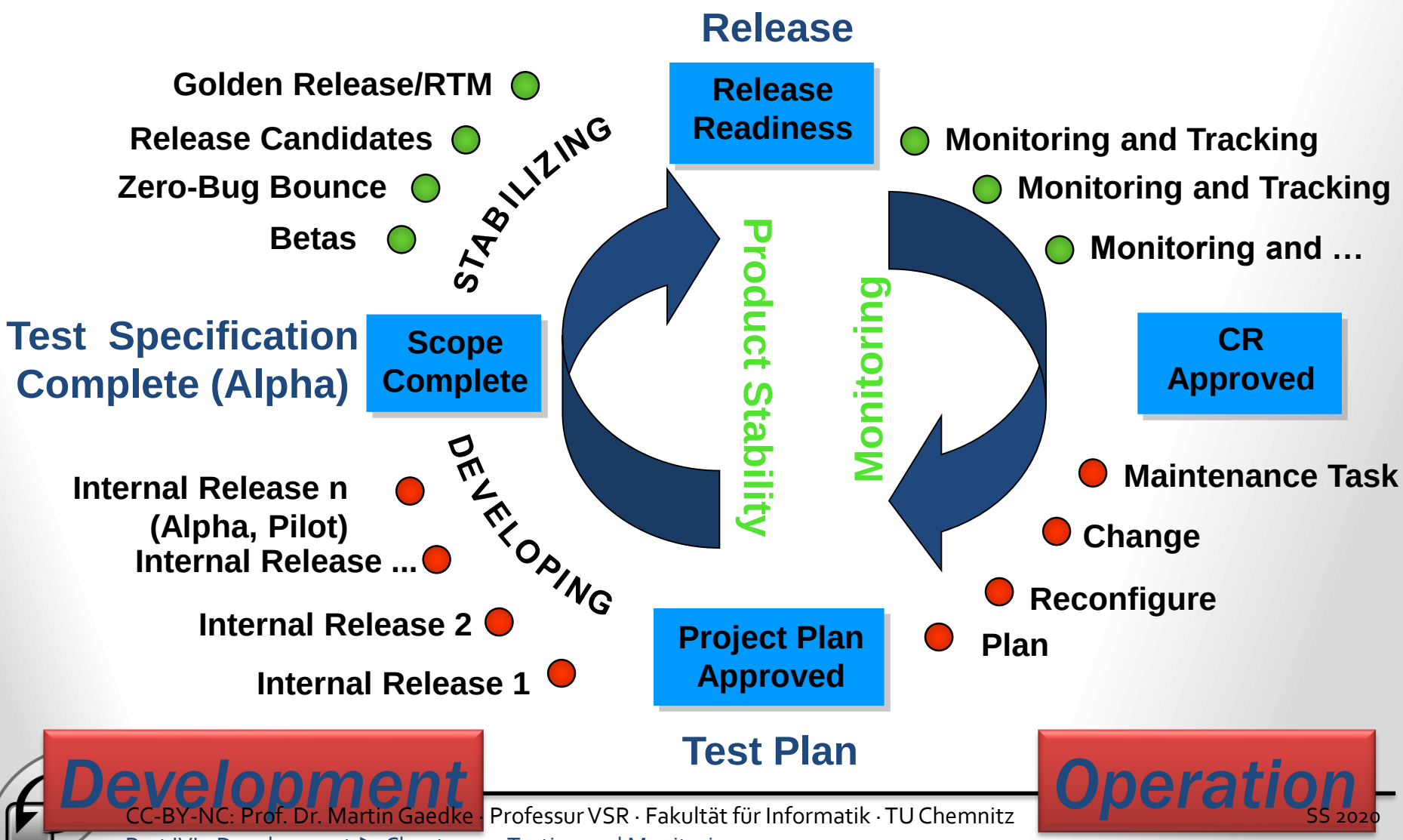
Done

Screenshot: Microsoft Application Center Test

Heat Maps

picture from http://blumenthals.com/blog/

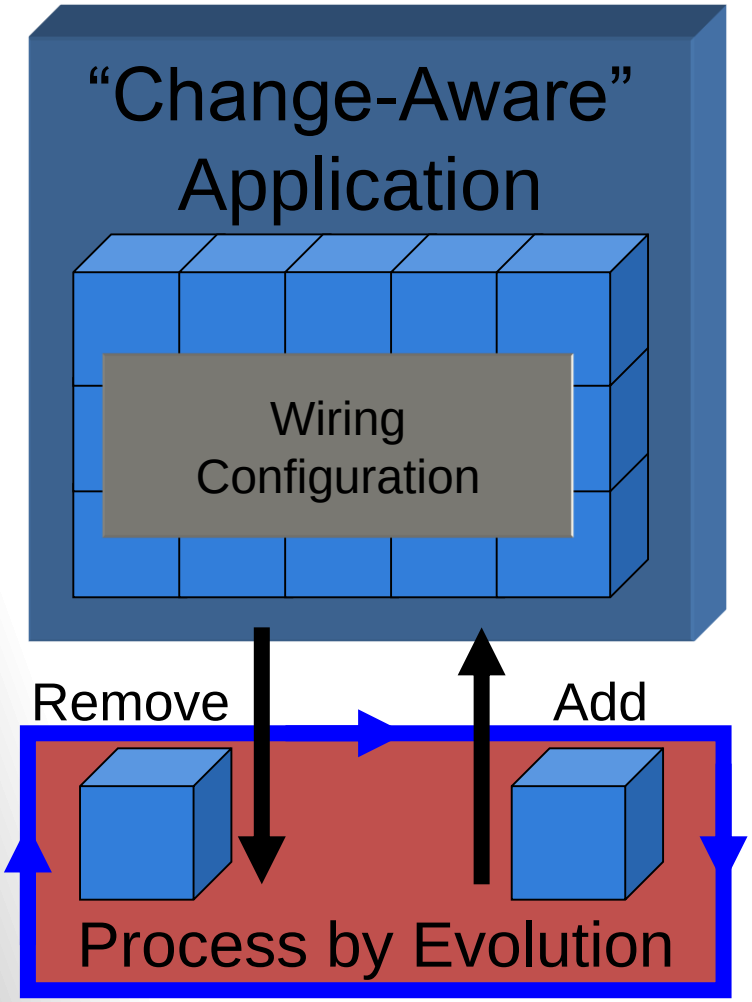
Evolution – Ongoing Process



CHAPTER://23

- Operation and Maintenance

Evolution: Plan For Change



- Evolution requires for continuous Operation and Maintenance, i.e.
  - Continuous testing and monitoring
  - Bugs or changed requirements → Change the application
  - Accepted changes (cf. Change Management and CCB) are handled
- Configuration as an approach to evolution
  - Add: Wire existing components by adding to configuration
  - Remove: Delete wiring context in component configuration

SECTION://1

- Availability

Achieve High Availability?

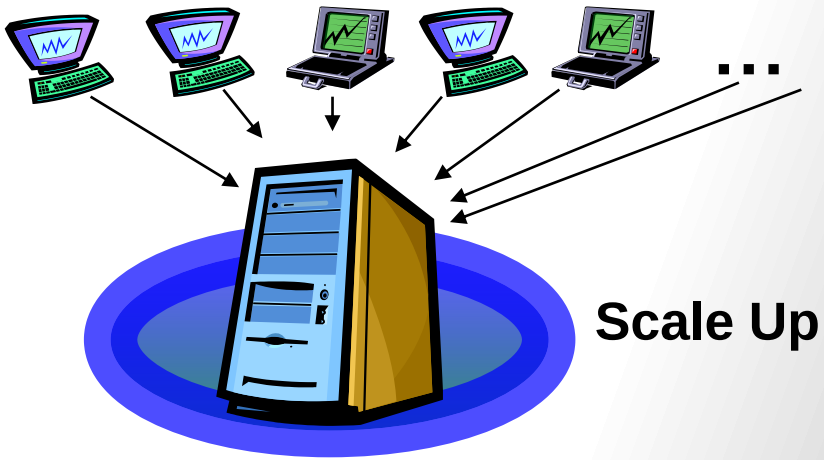
- It's deceptively simple …
  - Plan and prepare
- Key to high availability
  - Deploy systems to create **redundancy** – the key from a technology standpoint, e.g. replicate Web server application logic (scale out, DNS-round robin, Network Load Balancer)
  - Define processes for people to solve conflicts
  - Test, test, test
  - Monitor on a continuous basis

Technical Approach: NLB

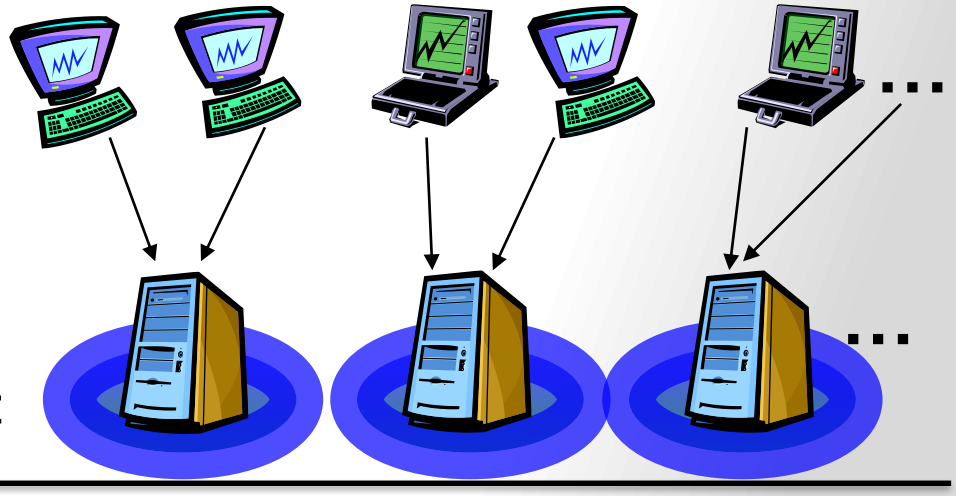
- NLB = Network Load Balancing
  - E.g. NLB-Service or Windows Load Balancing Service (WLBS)
- Generally used for scalability
- Can be used with databases
  - Front end switch for log shipping role change
  - Warm standby server
  - Protect analysis services

Considering Scalability (and the cloud)

- **Scale Up**: More "power" added to a machine
- **Scale Out**: The application logic unit is cloned across a set of identical servers

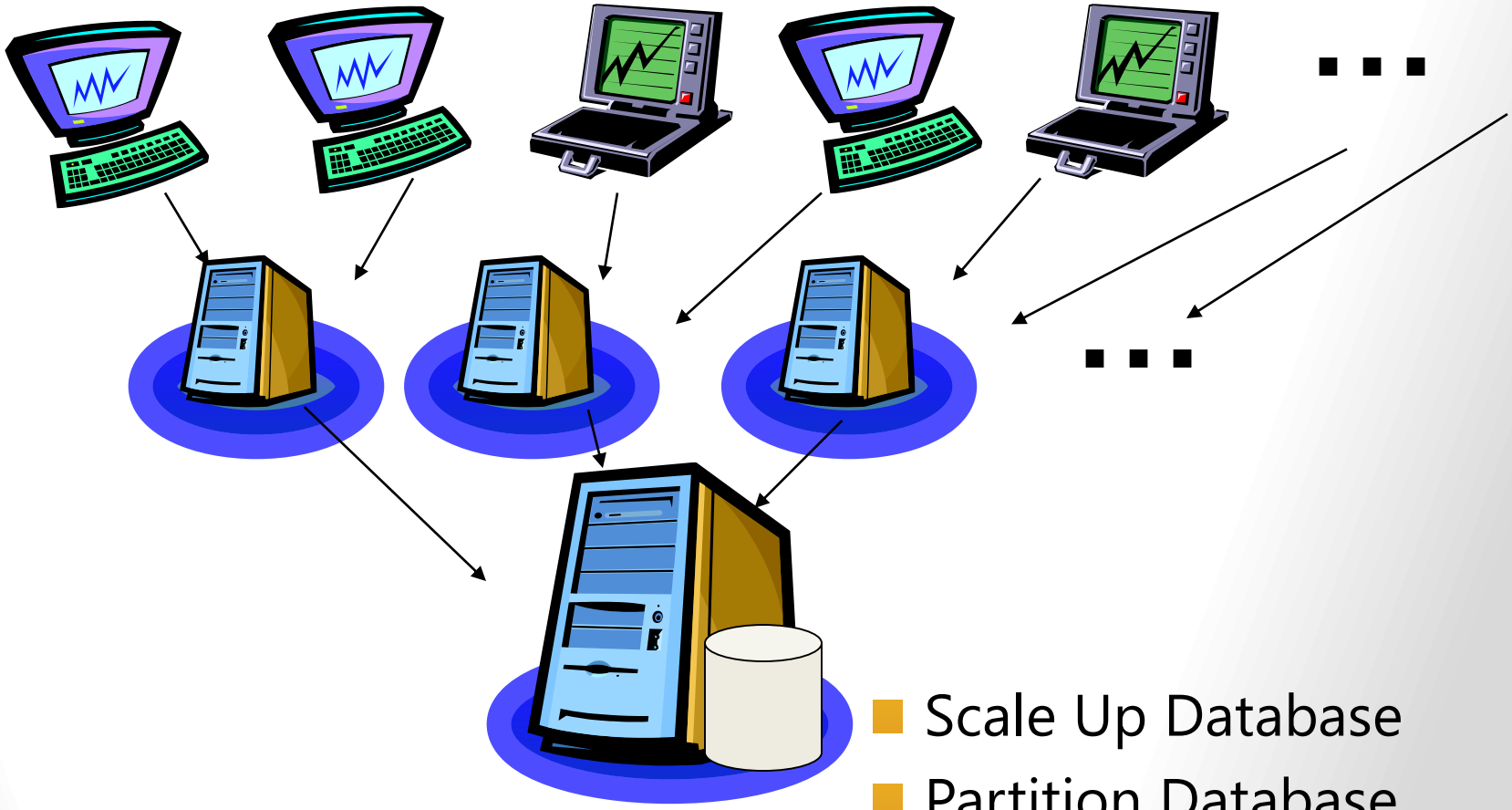


- How is scalability different in the context of the cloud compared to approaches mentioned above?



Scale-Out and Partition

- Scale out Web Servers and scale up Database



- Scale Up Database
- Partition Database

Partition Database

- Functional – Each functional area of a site gets its own database
  - Dedicated hardware to certain functions
  - Class of hardware per function
- Tables - Huge scale opportunity for large tables
  - Some modern database management systems provide special support for this
- Read-only Databases
  - Data changes do not occur often, e.g. Lecture Catalog
  - Use of Replicated Databases

Partition Web-Tier

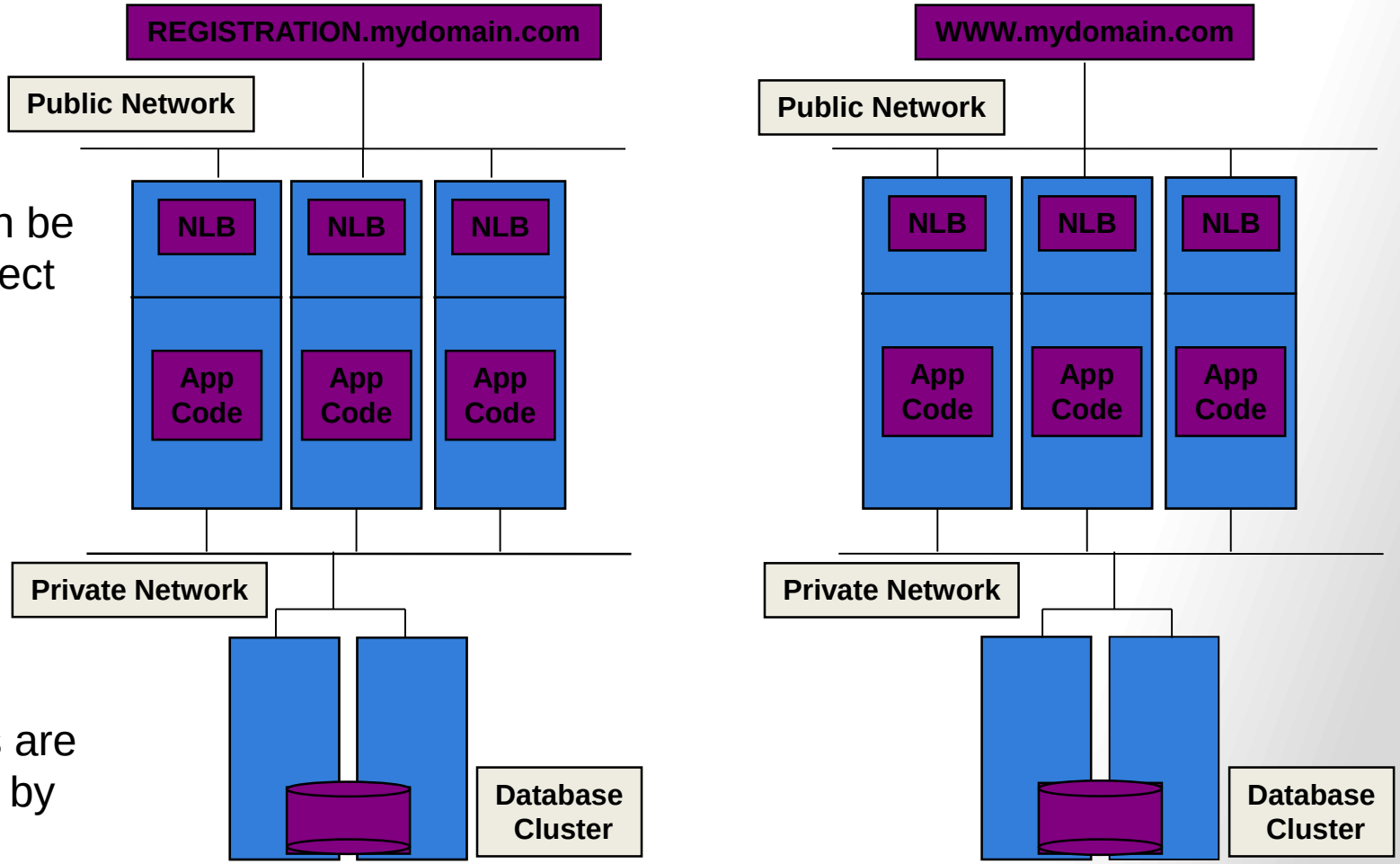
- Like Functional Database Partition
  - E.g. Partitions: *search*.Business.com, *www*.Business.com etc.

```
C:\WINNT\System32\cmd.exe - nslookup
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>nslookup
Default Server:  AP-Dr.Gaedke.local
Address:  10.0.0.254

> www.hp.com
Server:  AP-Dr.Gaedke.local
Address:  10.0.0.254

Non-authoritative answer:
Name:    www.hp.com
Addresses:  192.151.52.187, 192.151.53.86, 192.151.52.217, 192.6.118.97
          192.6.118.128

> www.ibm.com
```

- DNS Host Names or Hardware Solutions exist to distribute traffic to dedicated server/clusters
  - Simple Approach: DNS Round-Robin
  - E.g. www.myserver.com refers to several IP Addresses / physical servers

Scaleable Architecture

DNS Host Names can be used to direct Traffic to functional Clusters.



Databases are partitioned by Function.

Using Messaging Approaches

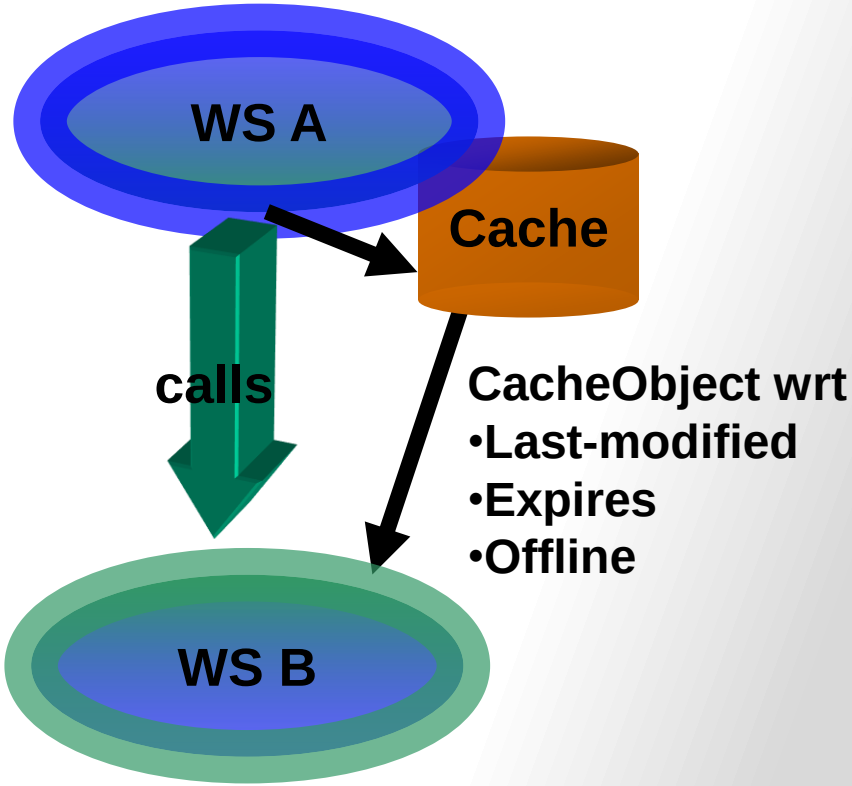
- Provide a high degree of scalability by decoupling the user experience from the backend processing
  - Asynchronous processing
- Example:
  - Order process consists of 3 stages (Producing, Packaging, Shipping) – usually takes some days
  - After ordering – User can check status of progress

Using Messaging Approaches

- Use queue- or port-oriented communication model where applicable
- Using asynchronous programming techniques whenever there are:
  - Opportunities for parallel processing
  - Batch type of operations
  - Interfacing with legacy applications
  - Real-time Operations

Considering Optimization

- Optimization ≠ Performance Tuning
- Reducing WS-Calls
  - Use caching or offline content generation
  - Check which navigation scenarios
- **Caching Approach**
  - "***Good Enough*** Hit" Function
  - Example: Olympics site



Example: Caching Candidates

SALT LAKE 2002
OFFICIAL SITE OF THE 2002 OLYMPIC WINTER GAMES

HOME | SITE MAP | PAGES EN FRANÇAIS

Qwest RIDE THE LIGHT™

Sports | Schedules | Athletes | Spectator Info | Shopping | Games Programs | Tickets

50 days until the Salt Lake Games

MOSTLY CLOUDY

Thursday, December 20, 2001
Current Salt Lake weather: 37°F(3°C)

Cheek's Week

- Speed skater sets American record, makes Olympic Team
- Witty's winning ways include new triumphs at U.S. trials
- Biathlon: Forsberg does it again

Todd Warshaw / Getty Images

Tickets

- 2002 Olympic Winter Games
- Official Olympic ticket auction
- Olympic Arts Festival
- 2002 Paralympic Winter Games

Get Involved

- Employment opportunities
- Team 2002 Online Community

Shopping

MSNBC msnbc.com msn
Official Online Content Providers

RIDE THE LIGHT™

NBCOLYMPICS.COM

Site presented by CHEVROLET Qwest GE

Done | Internet

Optimization

- Often Wiring Models include dynamic or transactional relationships
- Integration – Still a bit of black magic or "Art and Experience"
- Different approaches exist
  - ► Model dependent optimization is possible

SECTION://2

- Maintenance

Maintenance

- Web applications are like "living entities"
  - Like a Garden: Must be maintained to look nice.
- **Maintenance is any event that yields to a new iteration of the life cycle** (of a feature or the application as a whole)
  - **Often you will find: Maintenance** – Any development activity performed to modify or fix the Web application after it has been completed or reached some final milestone. Be aware of ad-hoc maintenance (code-and-fix approach)!
- Reasons for Maintenance
  - CR: Content, delivery / access, functionality
  - Maintenance is good – it is the beginning of evolution!

Aspects of Maintenance - I

- Content Maintenance (Main Activities)
  - Note: Never work on a live site! Use a staging approach
  - E.g. Content stored in Database and is easily manipulated using dedicated Editors. Reviewed Content is updated on Production Server
- Delivery Maintenance (Success Disaster)
  - "Perfect" Web application fails – if traffic increases dramatically
  - Prepare for scalability and availability

Aspects of Maintenance - II

- Functionality Maintenance (Crisis Management)
  - **Corrective** – Activities to fix application bugs and design flaws
  - **Adaptive** – Activities to make application work for a "Problem-Browser" configuration
  - **Perfective** – Activities to increase functionality (feature additions)
- Note:
  - If too many Bugs hinder the functionality and can not be solved in a few minutes:
  - **Provide a currently under maintenance page**

Tools for Maintenance

- Monitoring
  - Web application statistics terminology
  - Logs of server, router, etc.
  - Use: Log Analysis Software
  - Your own unit tests
- Feedback Channels
  - E.g. contact information, forms for user feedback etc.

CHAPTER://24

- Further Readings

Literature

- Chapter 8: Thomas A. Powell, Web Site Engineering, Prentice Hall PTR
- Chapter 19-24: Ian Sommerville, Software Engineering, Addison-Wesley
- Chapter 11: I. Jacobson, G. Booch, J. Rumbaugh, The Unified Software Development Process, Addison-Wesley, 1999

==========

Further information available at Lecture Web Site

==========

Important Links

- IBM's Ease of Use
  - http://www-3.ibm.com/ibm/easy/eou_ext.nsf/publish/558
  - http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/609
- Jakob Nielsen's web usability website
  - http://www.useit.com/alertbox/
- Microsoft Usability Home Page
  - http://www.microsoft.com/usability/
- Sun's Usability Testing of Web Concepts:
  - http://www.sun.com/980113/sunonnet/concepts.html
- User Interface Engineering
  - http://world.std.com/~uieweb

Danke für Ihre Aufmerksamkeit!

Gibt es Fragen!