



# Content Management in the Cloud: A Case of Interoperability of Services Across Organizational Boundaries.

Hans-Werner Sehring, T-Systems Multimedia Solutions, Germany

Computation World 2013, 31 May 2013, Valencia, Spain



# Outline.

- **Business Perspective.**
- Some Words About my Company.
- (T-Systems) Cloud Business.
  
- **(Personal) Research Perspective.**
- Content Management Systems (CMSs) as Interesting Service Offers.
- Pitfalls in and Approaches to an SaaS Content Management Solution.
- The CCMS Architecture as a Basis for Cloud-optimized CMSs.
- Summary and Future Work.



# Some Words About my Company.

# Deutsche Telekom. Partner for connected life and work.

Deutsche Telekom delivers one-stop services and solutions:  
for all customer communications needs – at home, on the move and at work.

**T-Mobile**



T-Mobile offers cell-phone solutions in the Netherlands, Austria, the Czech Republic, and the USA.

.....T-Mobile.....

**Telekom**



The Telekom subsidiaries provide products and services for the fixed network, mobile communications, the Internet and IPTV in Europe.

.....T.....

**T-Systems**



T-Systems delivers ICT solutions for major corporations and public-sector organizations worldwide.

.....T-Systems.....



# Deutsche Telekom. Facts and figures.

## We make the right connections.

- **Market position.** No. 1 on German market (phone, mobile, IT)
- **International.** Presence in around 50 countries worldwide
- **Revenues.** €58.7 billion, more than 50% generated outside Germany
- **Employees.** 235,000 worldwide, approx. 50% outside Germany
- **Customers/connections.** Over 180 million (broadband Internet, phone, mobile, IPTV)

## We run our business responsibly.

- **Data privacy.** A specially created Board area promotes the exchange of ideas with leading data privacy experts from the worlds of politics, education and business
- **Eco-friendliness.** Driving force for sustainable climate protection through the development of energy-efficient ICT solutions, both for ourselves and our customers
- **CSR.** One of the largest providers of vocational training in Germany with 10,000 trainees

## We think ahead.

- **Innovation and cooperation.** Member of over 50 international committees and more than 150 work groups
- **Deutsche Telekom Laboratories.** The company operates an R&D center affiliated to the Technical University of Berlin where 125 staff and 65 researchers from all over the world work in the field of IT and telecommunications
- **Deutsche Telekom Laboratories at Ben Gurion University.** Performs research into security solutions for telecommunications networks

Deutsche Telekom – an international market leader for connected life and work.



# T-Systems – Deutsche Telekom's subsidiary for major corporations. A global player with global resources.

**International presence:**  
Offices in over 20 countries, global delivery capability

**Focus:**  
Large corporations, multi-national companies, public-sector and healthcare organizations

**Market position:**  
No. 1 ICT provider in Germany  
No. 1 ICT provider for systems integration in Germany  
No. 1 ICT provider for the automotive industry in Western Europe

**Employees:**  
Approx. 48,200  
(25,500 in Germany, 22,700 outside Germany)

(2011 figures)

**Revenues:**  
€9.2 billion

(2011 figures)

•• **T** •• **Systems** ••

# T-Systems Multimedia Solutions.

- 1250 employees.
- 7 offices.
- €128 mio. revenues.
- 17 years of experience.
- 10 times in row we're said to be Number One\*.

\*Internet agency ranking of Germany's Bundesverband Digitale Wirtschaft (BVDW).

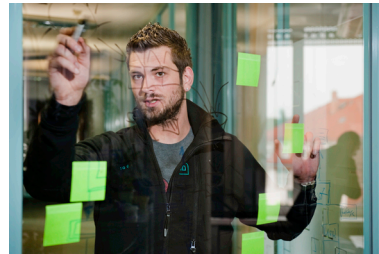


# Our Portfolio.



## Consult.

- Strategy Development
- Conception
- Consulting



## Create.

- Software & Integration
- User Experience
- Marketing



## Care.

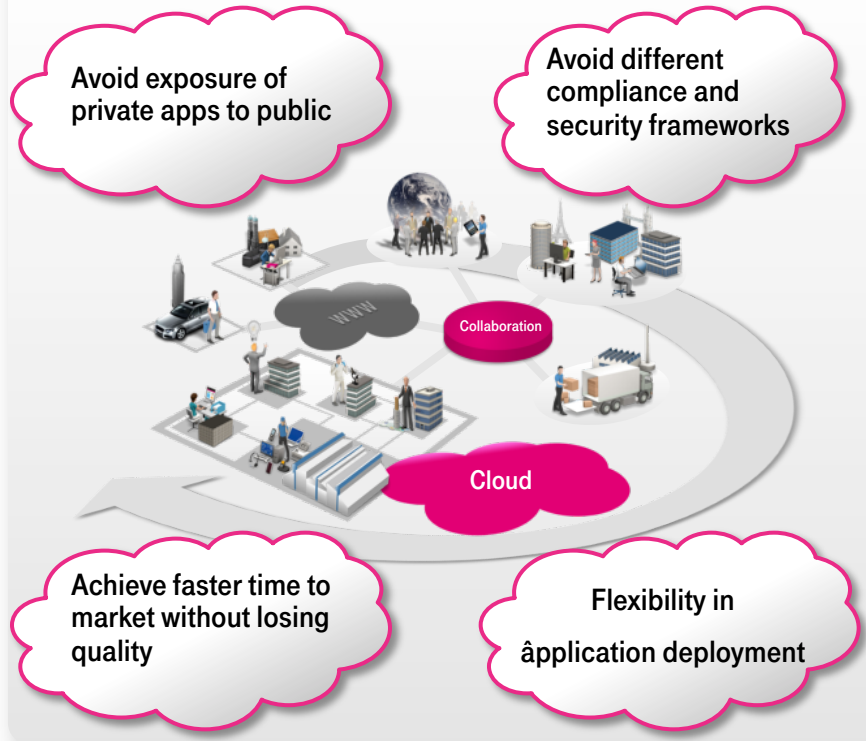
- Customer
- Content
- Service



# (T-Systems) Cloud Business.

# We support our Customers to bring Enterprise Applications into the Cloud ...

## Challenges in Application Operation



## Customer's CMO

- Up to 500 applications
- 50 – 70 % are developed in-house
- 30 % are standard applications
- Complex Application Life Cycle Management
- High costs for IT department
- 80 % of IT budget spend on operations
- Only 20 % is left for Innovation
- Strong need on application landscape consolidation
- Outsourcing

▶ Enterprises need a “One-Stop-Shop” for Business Applications.



# ENTERPRISES ARE BECOMING SMART SHOPPERS. JUST LIKE CONSUMERS.

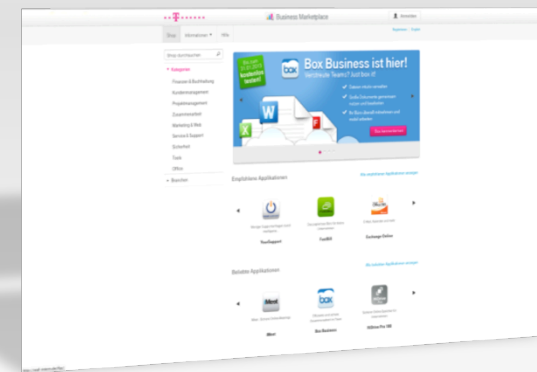
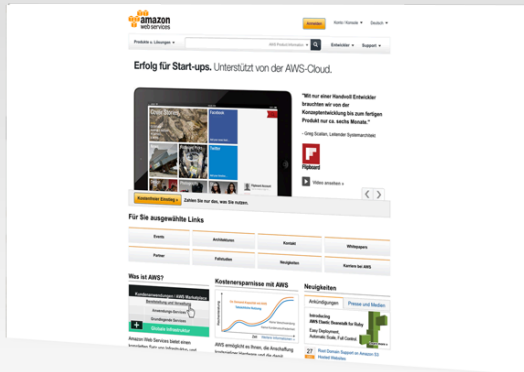
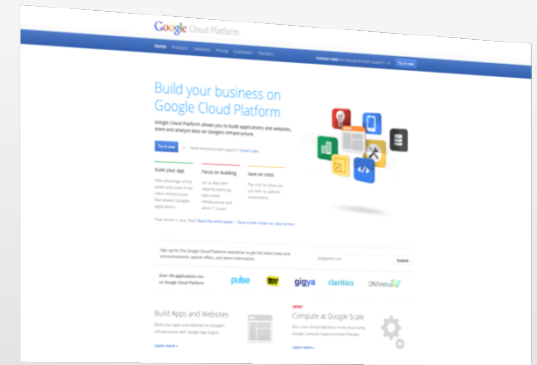
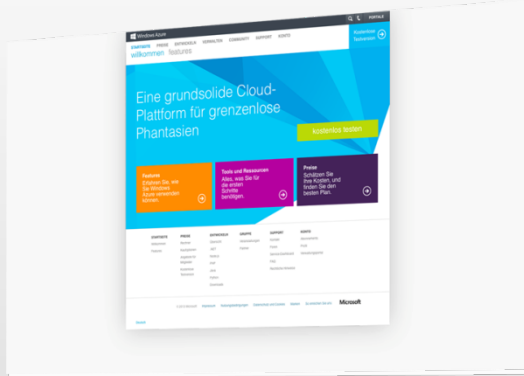
## ANALYSTS PREDICT:



“With **enterprise app stores**, the role of IT shifts from that of a centralized planner to a market manager providing governance and **brokerage services** to users.” (Gartner)

“**Cloud marketplaces** are a logical further development of cloud computing. Cloud marketplaces grant access to **IaaS resources, OS images** (DSI2.0) and other **software solutions** (Application Store).” (clouduser.de)

# APP STORES ARE NOTHING NEW. EVEN ENTERPRISE ONES. BUT WHAT ABOUT COMPANY-SPECIFIC APPLICATIONS?



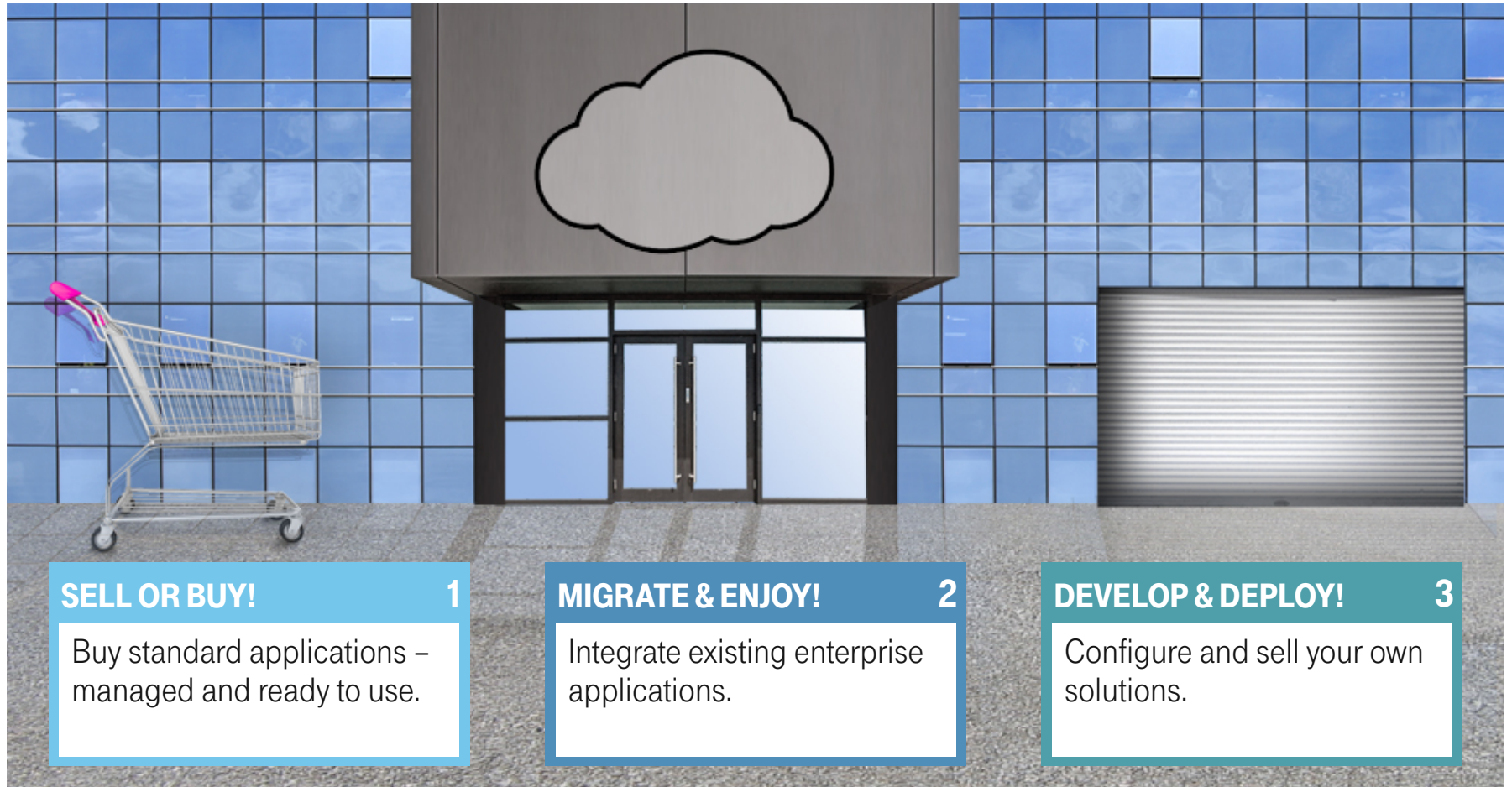
## NOW AVAILABLE:

Company-specific applications.  
and opportunities for developers.

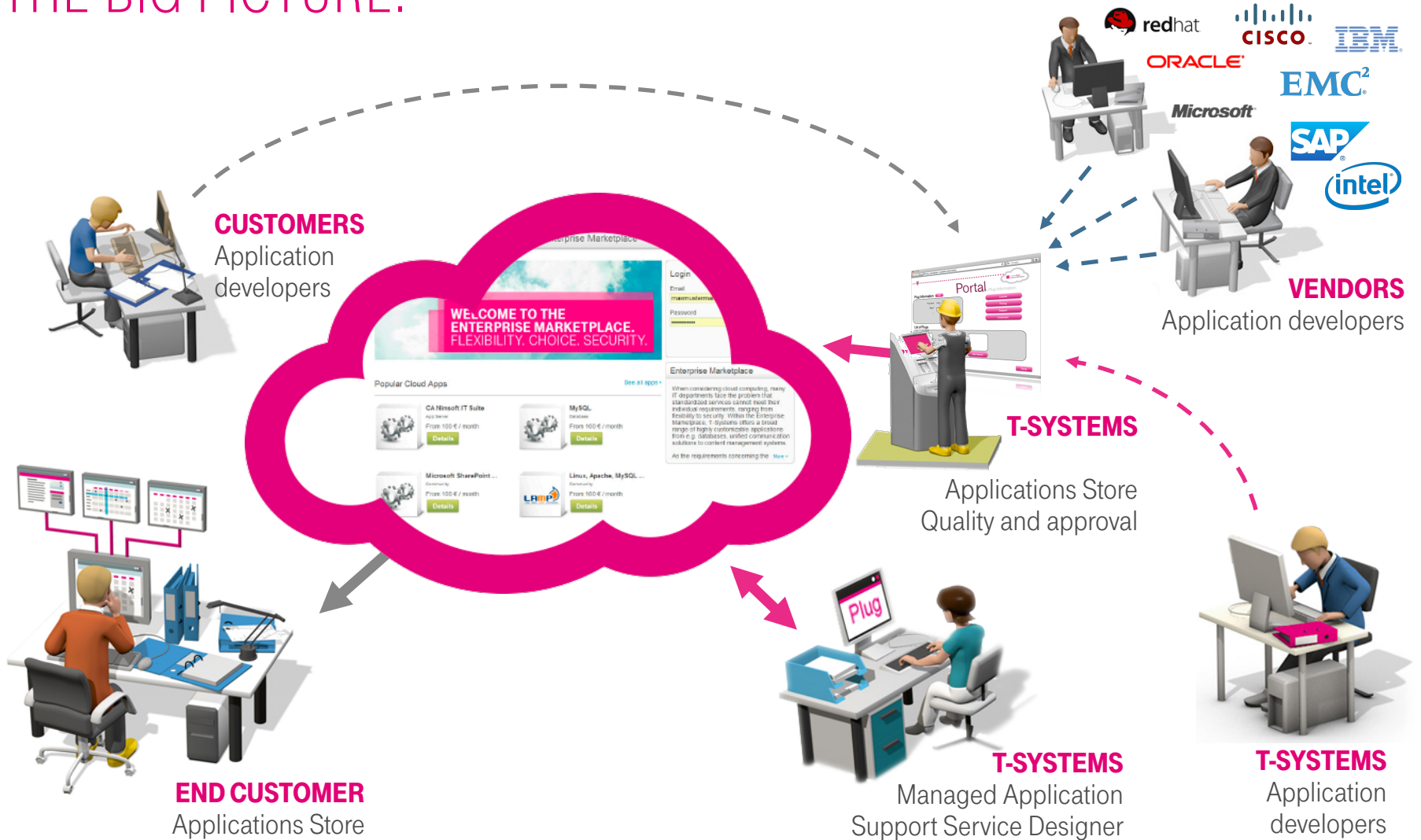


**T · · Systems ·**

# ENTERPRISE MARKETPLACE. THREE BENEFITS IN ONE.



# ENTERPRISE MARKETPLACE. THE BIG PICTURE.



T · · Systems ·

# OUTLINE.

- **Business Perspective.**
- Some Words About my Company.
- (T-Systems) Cloud Business.
  
- **(Personal) Research Perspective.**
- Content Management Systems (CMSs) as Interesting Service Offers.
- Pitfalls in and Approaches to an SaaS Content Management Solution.
- The CCMS Architecture as a Basis for Cloud-optimized CMSs.
- Summary and Future Work.



# Content Management Systems as Interesting Service Offers.

(individual research from here on)

# What Makes a CMS an Interesting Outsourcing Target.

- Possible **savings** (CAPEX, OPEX) when digital content does not constitute the primary business objects.
- **Elasticity** allows reaction to load peaks (unevenly distributed business, special events).
- Playout components can move closer to the customer ('s network) for higher **performance**.
- **Availability** by replication (see “Amazon EC2 failure”), software + content.
- Problem of vendor lock-in shifted towards service provider who takes the **risk** here.
- **Outsourcing**: full service offerings (conception, graphics design, ..., support).
- Easy (little investment needed) evaluation of **tools**, e.g., for marketing automation.
- Content producers, providers, and enhancers are often **distributed** anyways.

Typical Cloud advantages – savings, direct accountability, elasticity, risk, ...  
(plus some more) – can be seen here.

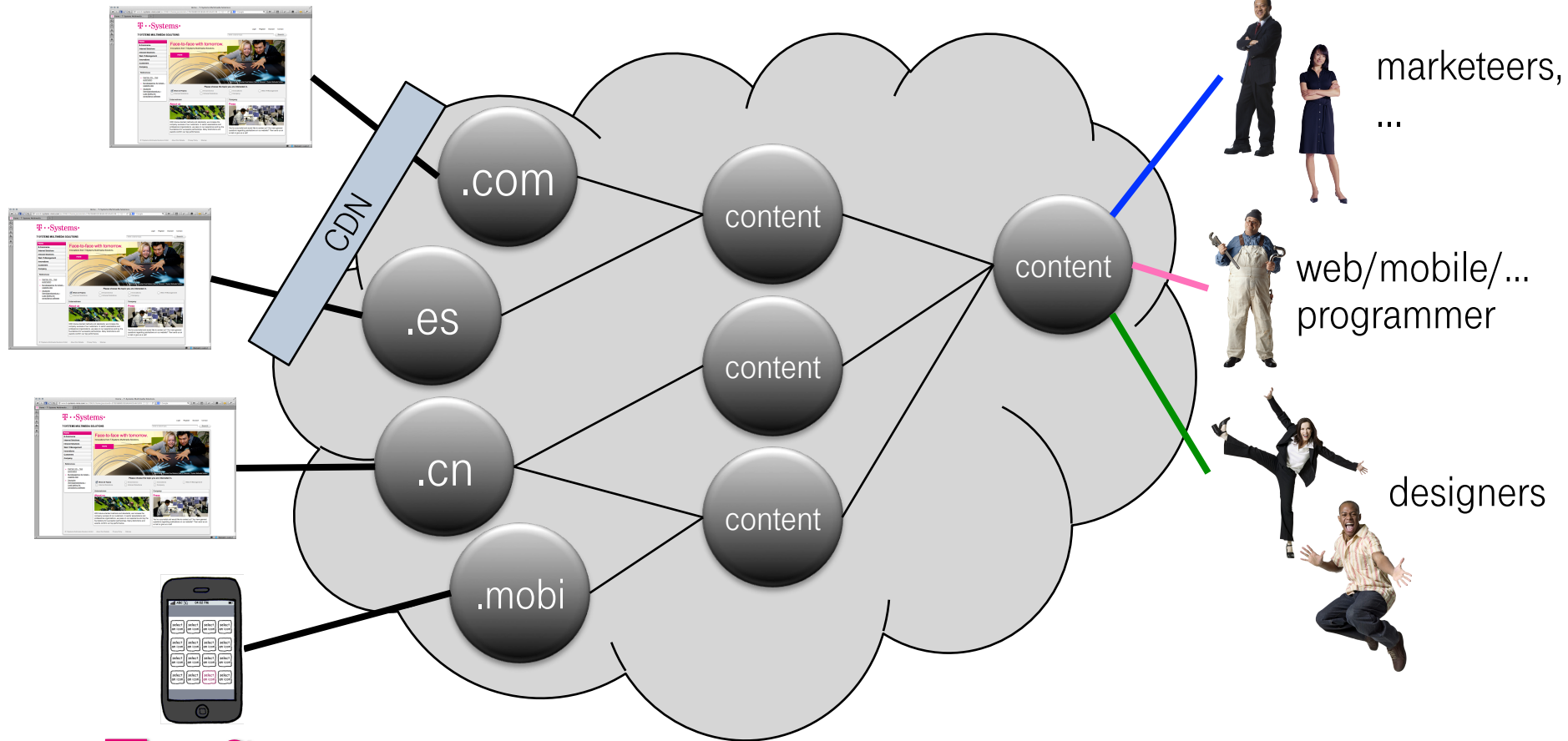
# What Makes a CMS an Interesting Outsourcing Target. Elasticity.

- Reaction to load peaks by dynamic **horizontal scalability**.
  - Powerful CMSs can run in cluster mode.
  - Simple payout (“Web 1.0”) scales well.
  - Interactive applications (“Web 2.0”, user-generated content, intranet, extranet) need suitable architecture.
    - Possibility to segment by tenant, language, publication channel, ...
- **Vertical scalability** by ...
- ... dedicating more / more powerful resources.
- ... moving content between CMS products usually is **non-dynamic**.
  - Specific functionality, platforms cannot substitute each other.
  - Proprietary interfaces, non-portable code.
  - Different content models, even different modeling paradigms, content hard to migrate.



# What Makes a CMS an Interesting Outsourcing Target. Performance.

- Playout components can move closer to the customer ('s network) for higher **performance**.



# Pitfalls in and Approaches to an SaaS CMS Solution.

# Pitfalls in an SaaS CMS Solution.

## Security and Ease of Setup.

- **Integrated systems** usually have not been designed with (territorial) distribution in mind, in particular in the pre SOA area (but even then).
  - Strong **interconnections** between systems, e.g., CMS (as PIM) and shop, CMS and ERP system, ... to deliver overall functionality of, e.g., a shopping site.
  - With an off-the-premises Cloud: calls from “outside”, **vulnerability** of the backend.
  - Internally stacked **DMZs**, policy to only accept calls from components in the more secure to components in the less secure zone (plus authorized exceptions).
- **Content and its lifecycle:** Even if content is not the primary good, ...
  - ..., there often is a vast amount of it (bulk media data, revisions).  
Effort to import it into a content management service.
  - ..., it is of high value / expensive to re-create.  
Contractual and technical issues for re-export when changing service provider.



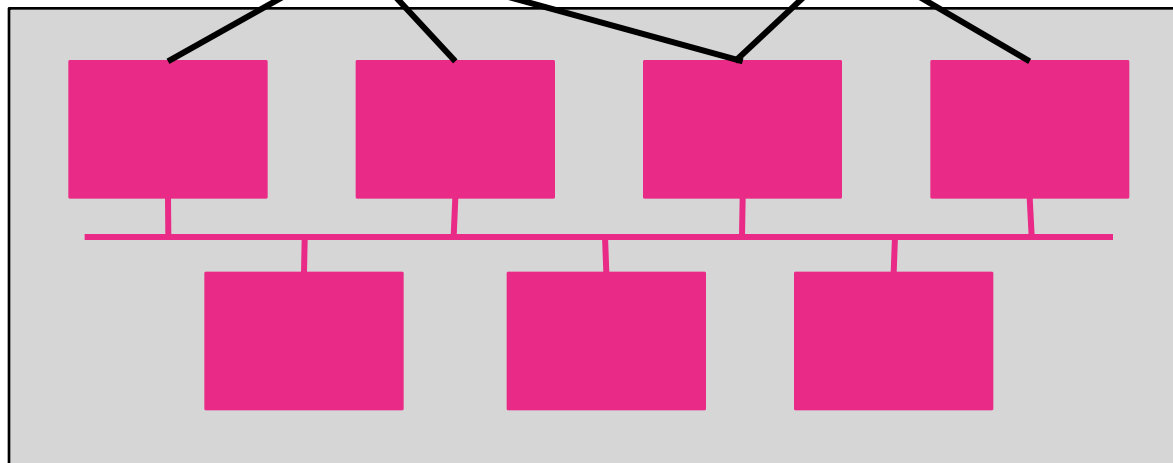
# Pitfalls in an SaaS CMS Solution. Typical Status Quo.

Internet DMZ in  
operations  
center



can be reached  
from Internet,  
may access  
backends

secure DMZ in  
operations  
center

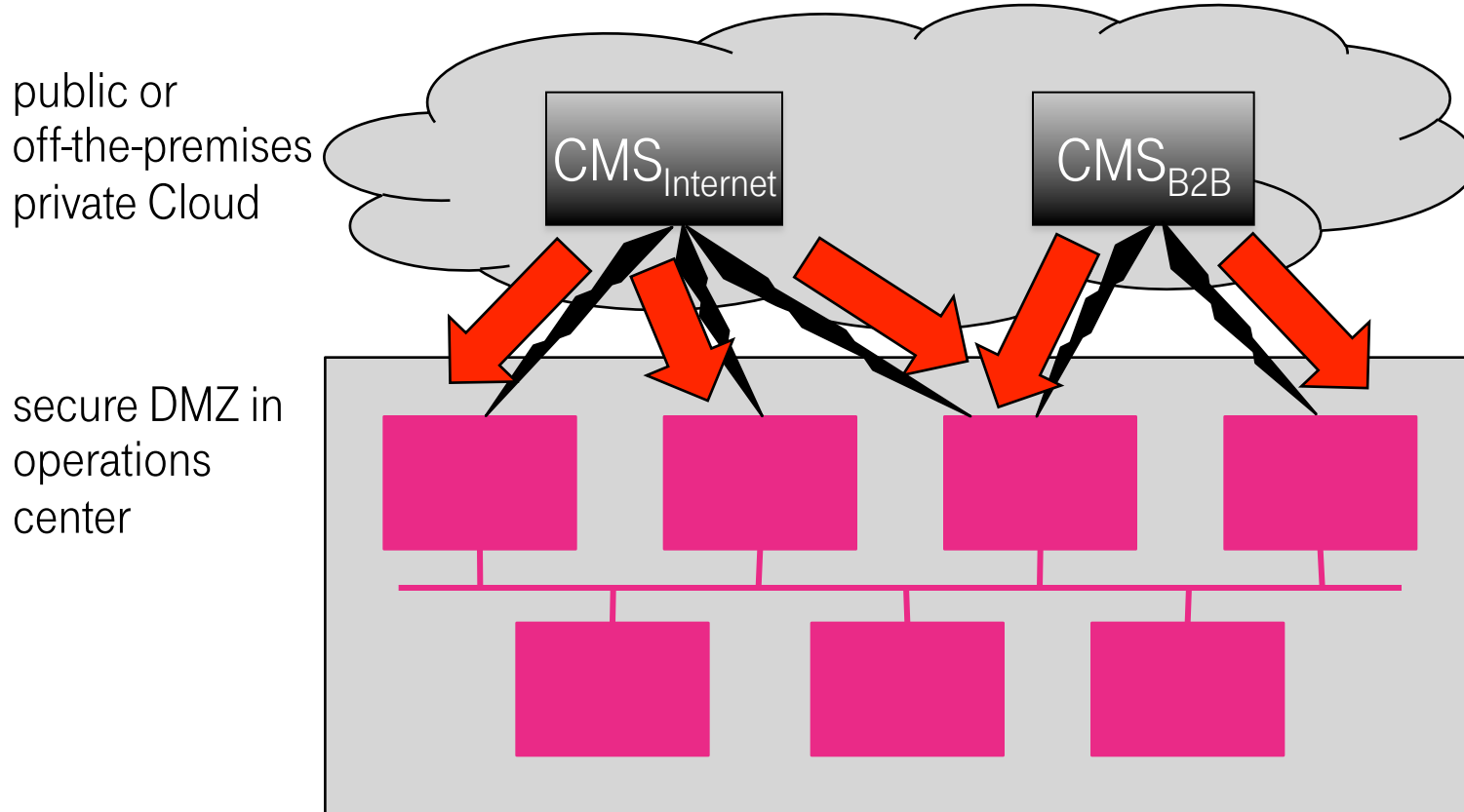


may be  
reached from  
CMSs in  
Internet zone



# Pitfalls in an SaaS CMS Solution.

## Moving CMSs into the Cloud – Forbidden Backend Calls.





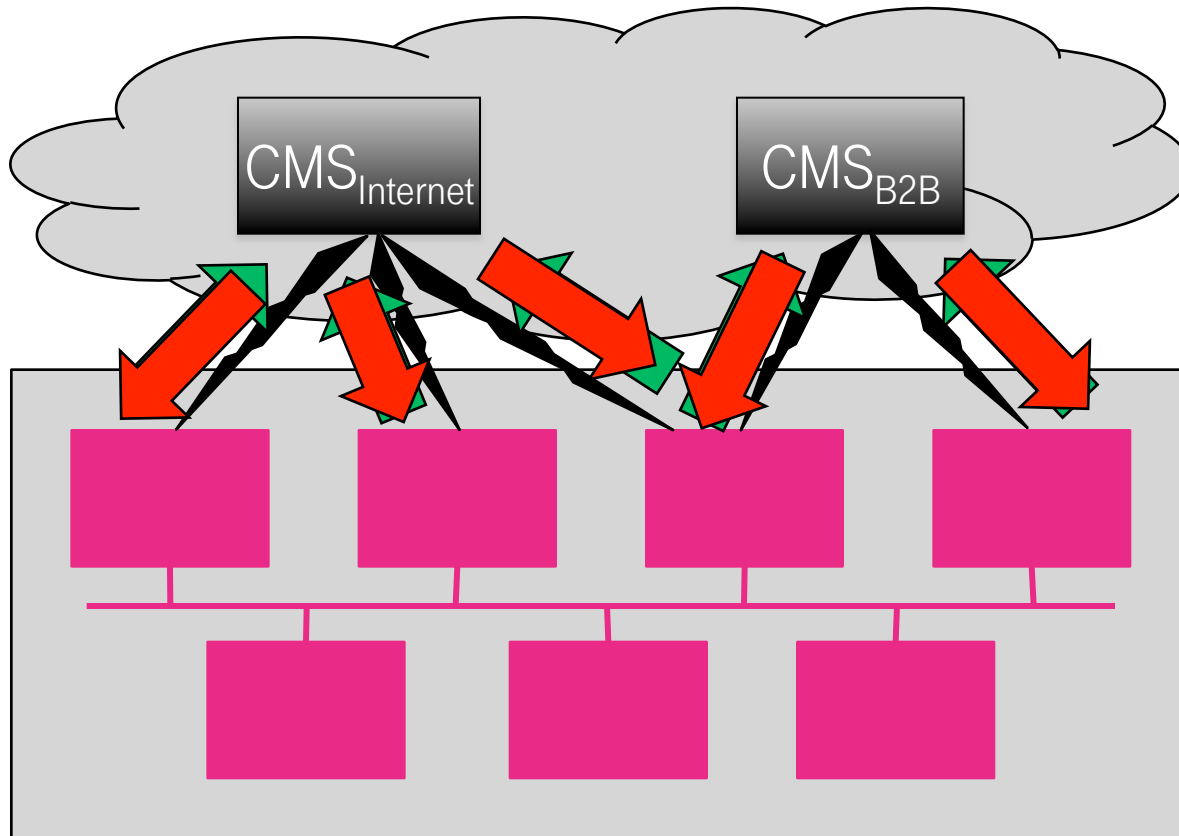
# Discussion of Approaches.

- **Avoid calls from cloud to systems in data center:**  
Calls from data center to cloud, not the other way round.
- **Regulate calls from cloud to backend systems:**  
Trustful communication.
- **Avoid physical barrier between internal and external systems** by means of a logical architecture:  
Non-monolithic CMS with local modules remotely connected to modules in a private cloud.



# Discussion of Approaches. Moving CMSs into the Cloud - Callbacks.

public or  
off-the-premises  
private Cloud



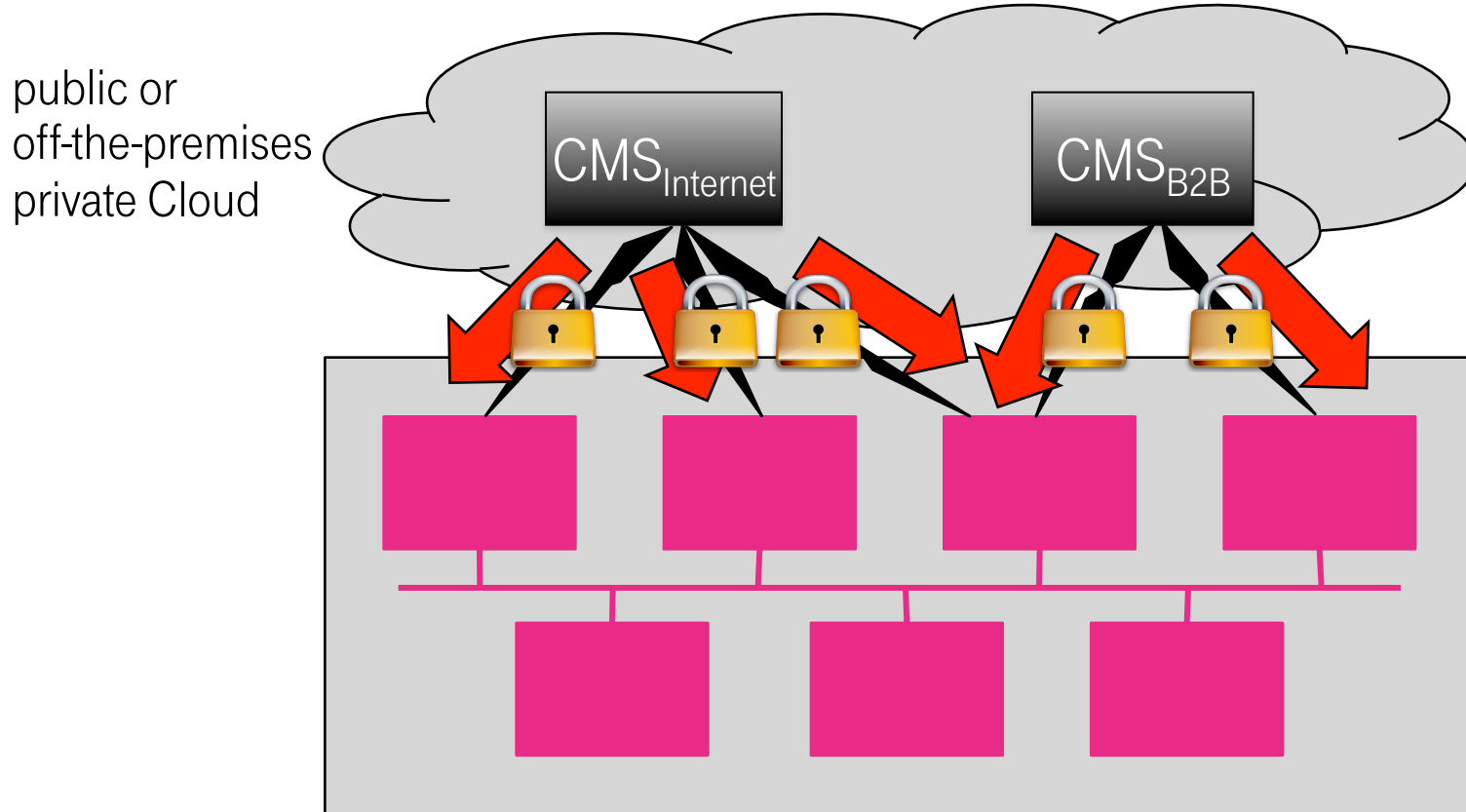
# Discussion of Approaches.

## Calls from Data Center to Cloud Only.

- Backend systems call CMS, need to address services in the Cloud.
- **Security issue:** CMS instances have to register observer with backend systems  
⇒ need to expose backend details.
- **Communication issues:** need for a standardized message-oriented middleware  
⇒ for all existing standards, implementations differ.  
Need for a standardized way of addressing content  
⇒ no standard that is implemented by all relevant CMSs.
- **Scalability issue:** backend systems need to maintain list of active CMS services (callbacks)  
⇒ obstacle to scalability.
- Not a viable approach.

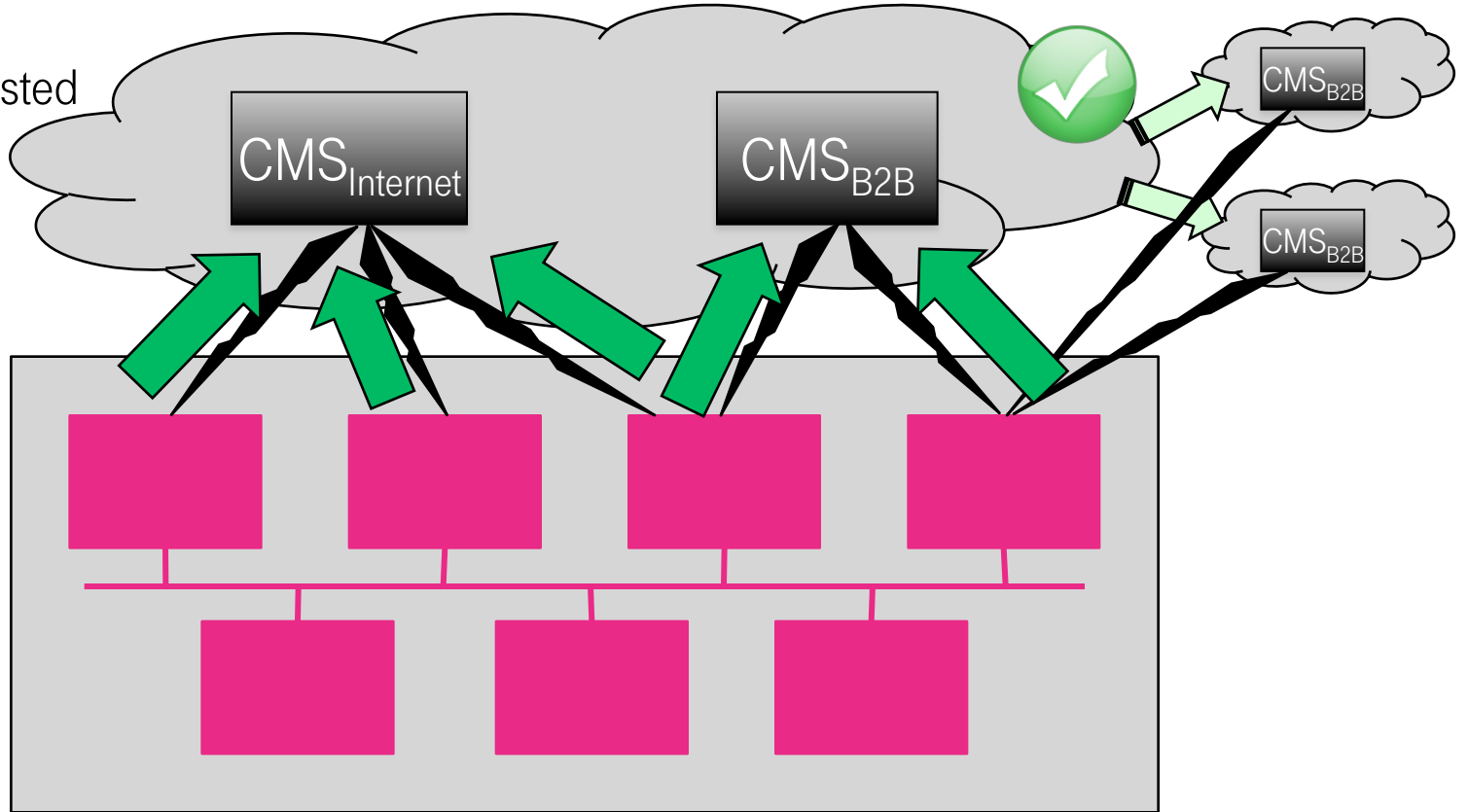


# Discussion of Approaches. Authentication by Providing Credentials or by Encrypted (Signed) Messages.



# Discussion of Approaches. Web of Trust.

private or well tested  
off-the-premises  
private Cloud



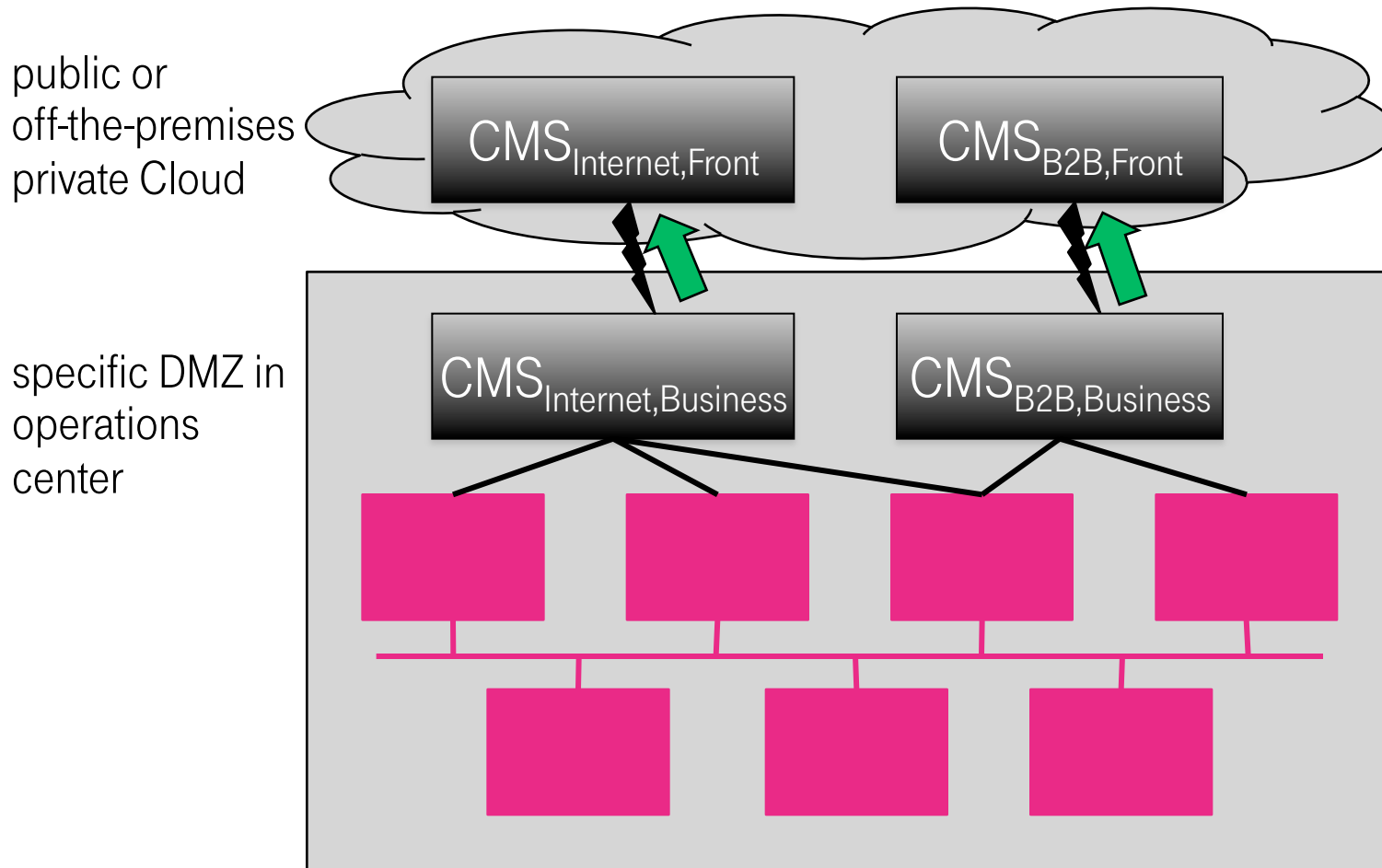
# Discussion of Approaches. Trustful Communication.

- (Machine) **Authentication:**
  - Opening systems to the Internet that were not exposed before ⇒ typically unwanted.
  - Credentials have to be delivered to Cloud ⇒ vulnerability to theft of these.
- **Tunnel- / VPN-based:**
  - Overhead for encryption ⇒ usually prohibitive.
  - Need to deliver certificates etc. to Cloud ⇒ vulnerability to theft.
- **Web of trust:**
  - Not strict enough for valuable backend resources, not testable/certifiable.
  - Not exactly reflected in Cloud infrastructure.
  - Literature: Security as a Service, Intrusion Detection as a Service (IDaaS), ...
- Not a viable approach.



# Discussion of Approaches.

## Breaking up CMSs, not Moving Whole Systems into the Cloud.



... **T** ... **Systems** ...





# The CCMS Architecture as a Basis for Cloud-optimized CMSs.

# The CCMS Architecture as a Basis for Cloud-optimized CMSs. Requirements to a Distributed CMS.

- Most promising of the three approaches: **non-monolithic CMS**.
  - Communication (logically) inside the distributed CMS component(s).
  - Trusted modules at each site, trusted communication.
- How to **deliver** modules to cloud?
  - Code generation from a platform-independent specification for a particular platform.
  - Instantiation of components for elasticity.
- Employ **model-driven approach** of CCMS.
  - Coherent content structures and defined communication paths through common high-level domain model.
  - Generation of software for different target platforms or language environments.
  - Architecture that supports evolution of content models without breaking communication paths.



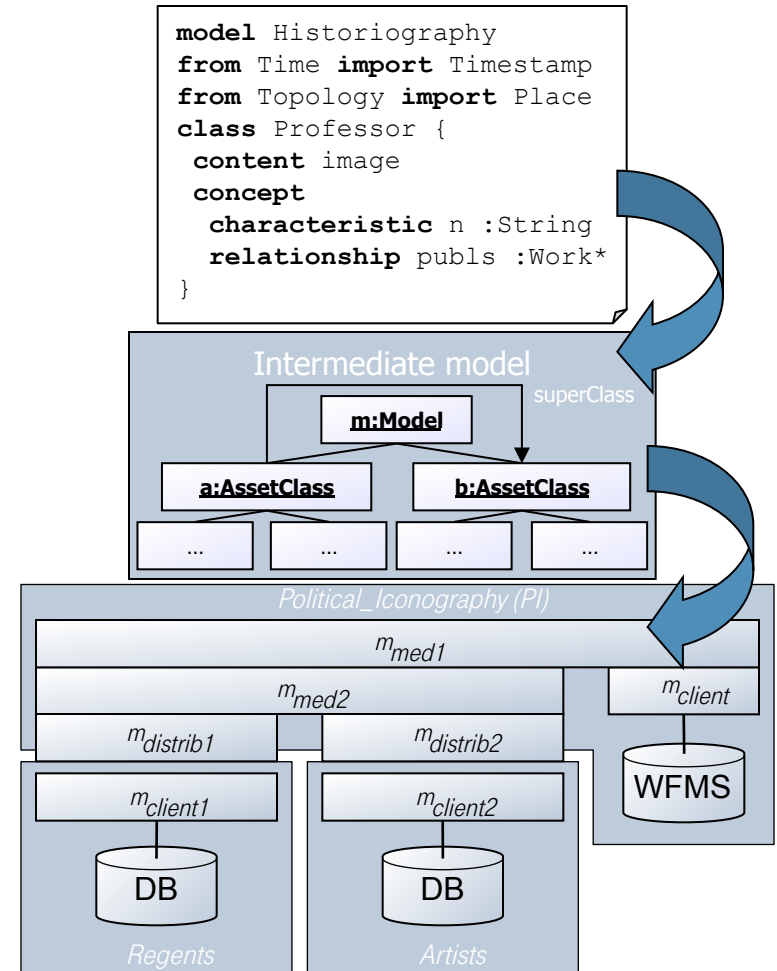
# CCMS Architecture as a Basis for Cloud-optimized CMSs. Creation of Heterogeneous Software Systems

- How to combine components that have been created by distinct generators?
- **General components:** generate independent components and ...
  - adapt the components [Assmann03, Brichau05]
  - create domain-specific wrappers [Gupta02]
  - create glue code connecting them [Assmann98]
- **(Web) Services:**
  - require common contracts (much more than interface!)
  - as part of that require common interpretation of parameters (content)
  - require correct representation/interpretation of content managed by all base CMSs
- **CCM systems:**
  - modules (not components) that are generated according to the same model
  - ⇒ coordination of the various generators



# CCMS Architecture as a Basis for Cloud-optimized CMSs. Concept-oriented Content Management Systems (CCMSs).

- **Technologies for adaptive systems:**
  - **modeling language**
    - modeling performed by domain expert
    - open for changes
  - **model-driven system development**
    - incremental generation
    - fully automatic, without developer intervention
    - preconfigured system patterns
  - **software architecture**
    - component based
    - evolution friendly

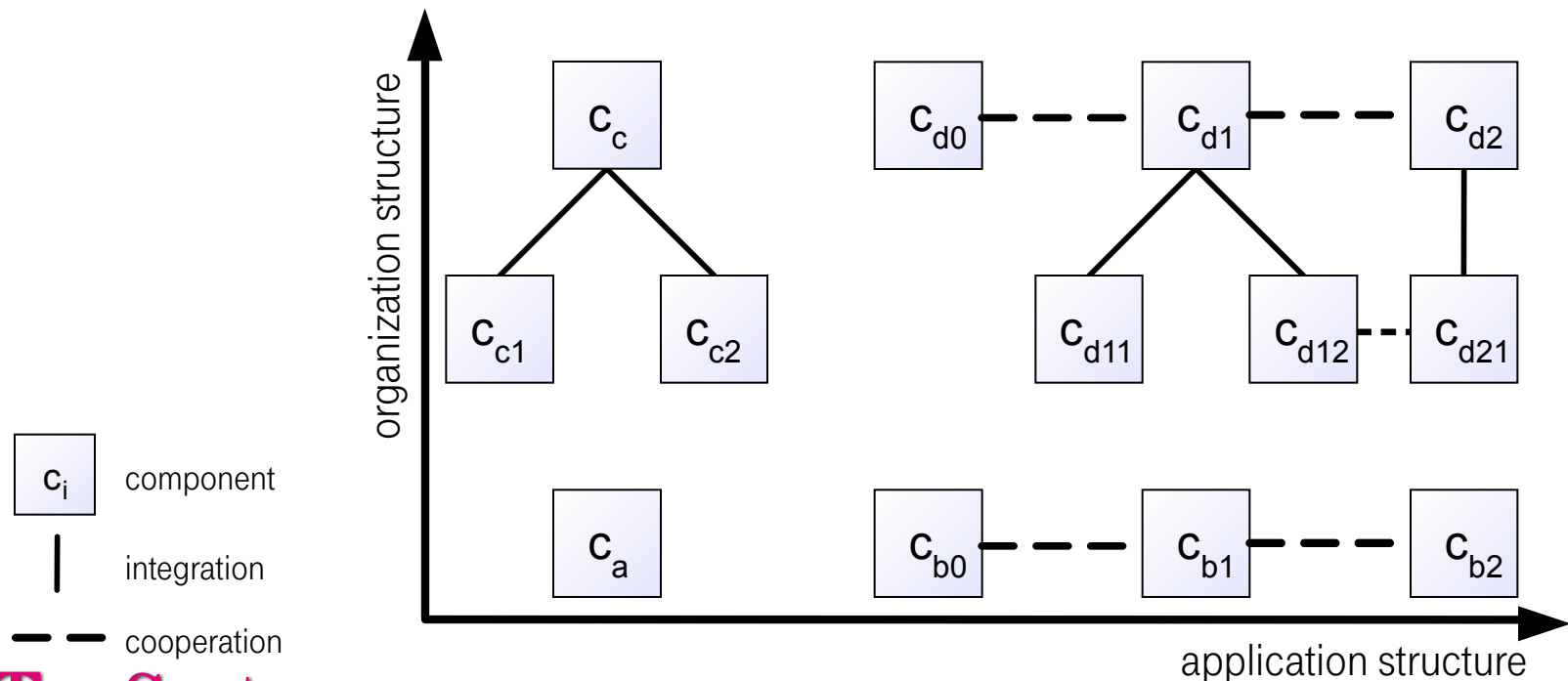


... **T** ... **Systems** ...

# CCMS Architecture as a Basis for Cloud-optimized CMSs.

## Logical CCM Architecture: Components

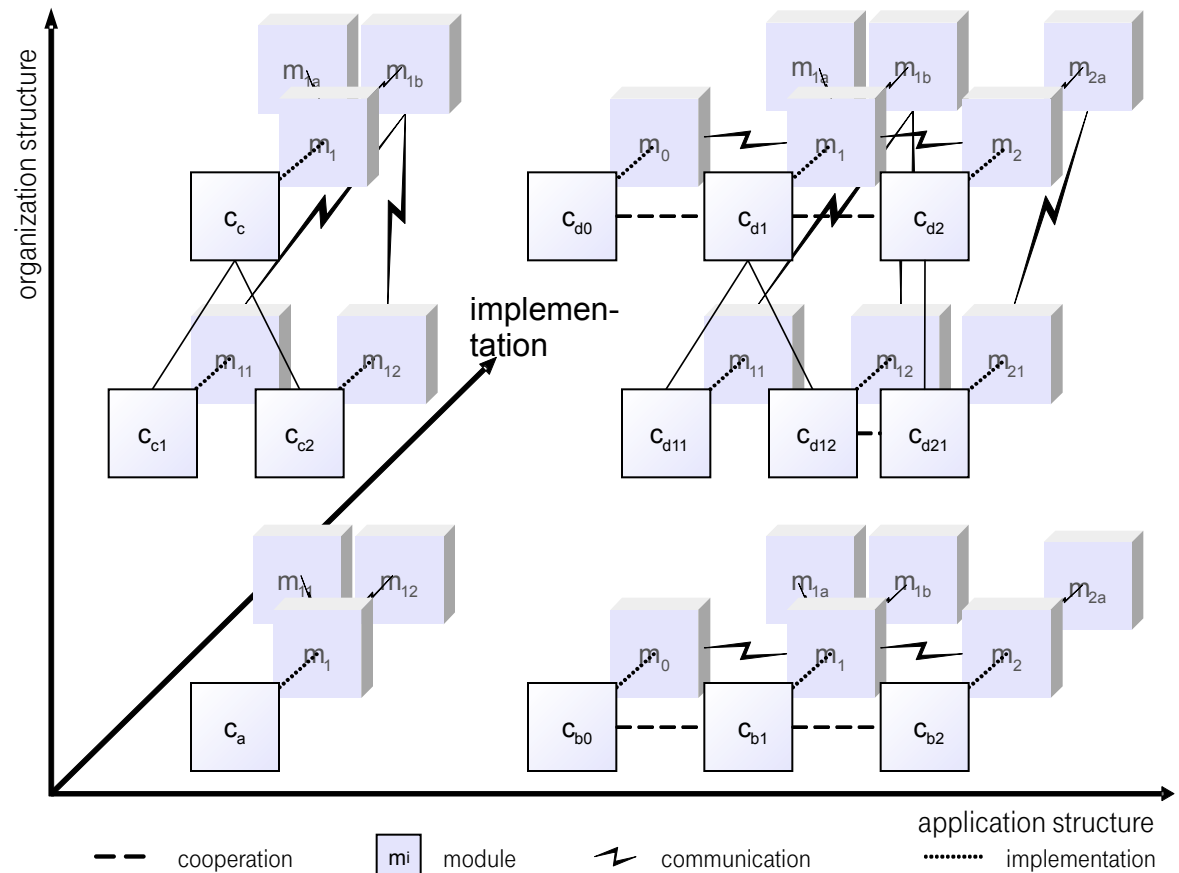
- CCMSs consist of **components**
  - one component for each domain model
  - cooperation for domain combinations
  - integration to model revisions and to derive variants



# CCMS Architecture as a Basis for Cloud-optimized CMSs.

## CCMS Implementation: Modules

- **Modules** implement components
- Reconfigurability at runtime for dynamics through:
  - *Separation of Concerns* by module kinds with distinguished functionality,
  - uniform module API, and
  - statelessness of modules.



# CCMS Architecture as a Basis for Cloud-optimized CMSs.

## Module kinds for CCMSs

- Module kinds for the construction of CCMSs:
  - **Client module:**  
Mapping to the generic concepts of a *base system*
  - **Server module:**  
Provision of functionality according to a standardized interface
  - **Coordination module:**  
Delegation of requests according to a certain strategy
  - **Adaptation module:**  
Conversion between different models
  - **Distribution module:**  
Connection to remote module of a (distributed) component



# CCMS Architecture as a Basis for Cloud-optimized CMSs. Analysis.

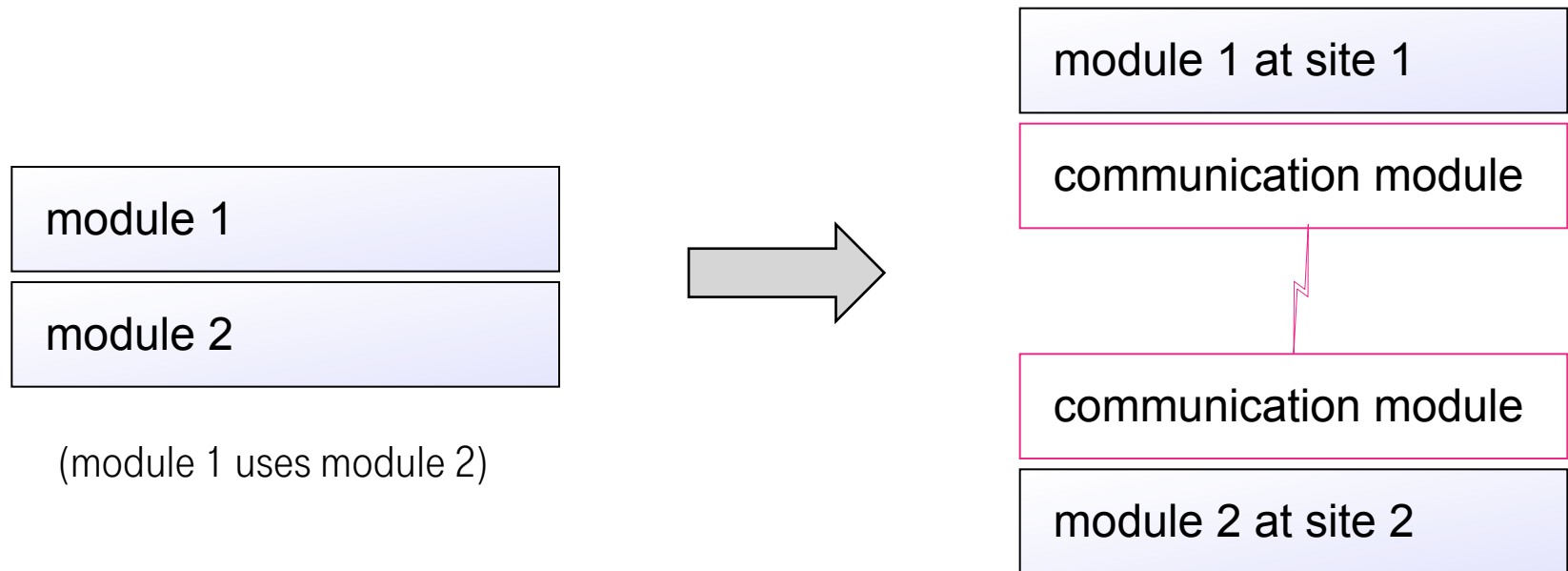
- **Pros:**
  - **Distribution:** naturally through module setups (collection of module instances).
  - **Accountability:** traffic in every module instance / module class can be measured.
  - **Elasticity:** through dynamic setup changes.
  - **Risk of vendor lock-in:** leveraged for both local middleware and Cloud environment.
- **Cons:**
  - Need to introduce CCMS architecture, enforce module API.
  - Cloud service provider has to be able to deploy and configure modules.
- **Arguable:**
  - “Programming” on the level of module configurations allows Cloud customers to customize software with less effort. But need to learn CCM.
  - Dependency on MDD can be neglected in this scenario. Dynamic systems evolution not necessary in all cases.





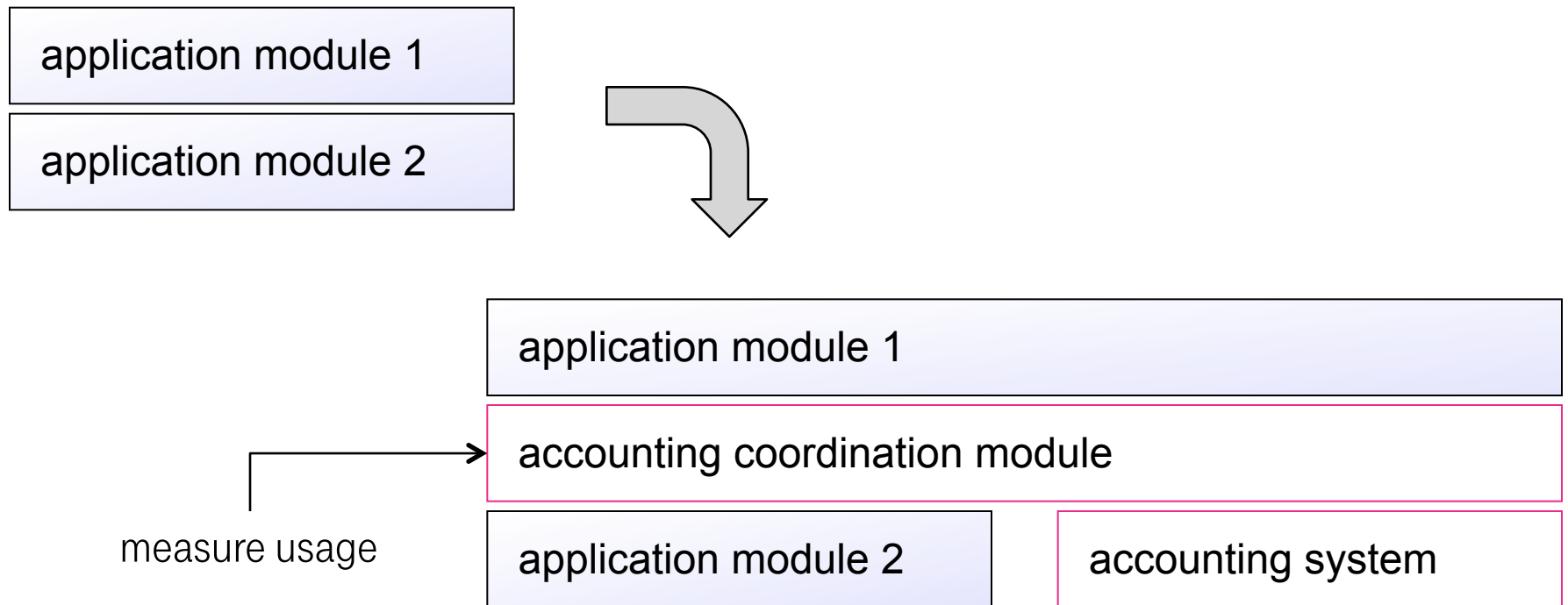
# CCMS Architecture as a Basis for Cloud-optimized CMSs. Distribution.

- **Distribution:** naturally through module setups (collection of module instances). So-called “communication modules” are generated by the model compiler and can connect any two modules.



# CCMS Architecture as a Basis for Cloud-optimized CMSs. Accountability.

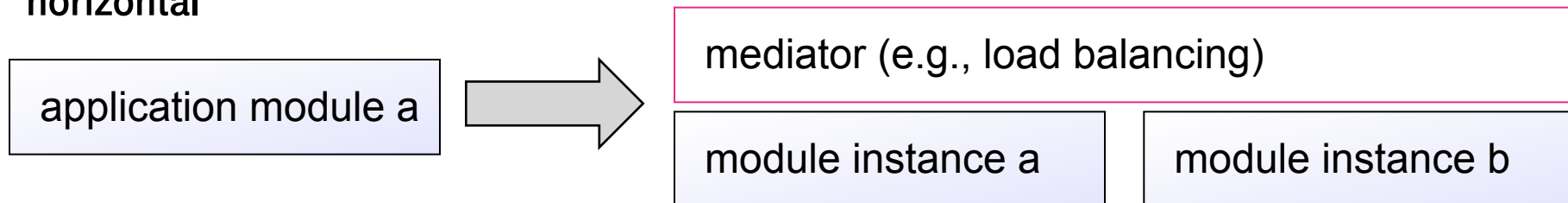
- **Accountability:** traffic in every module instance / module class can be measured. Allows measuring according to not only technical, but also application-specific metrics.



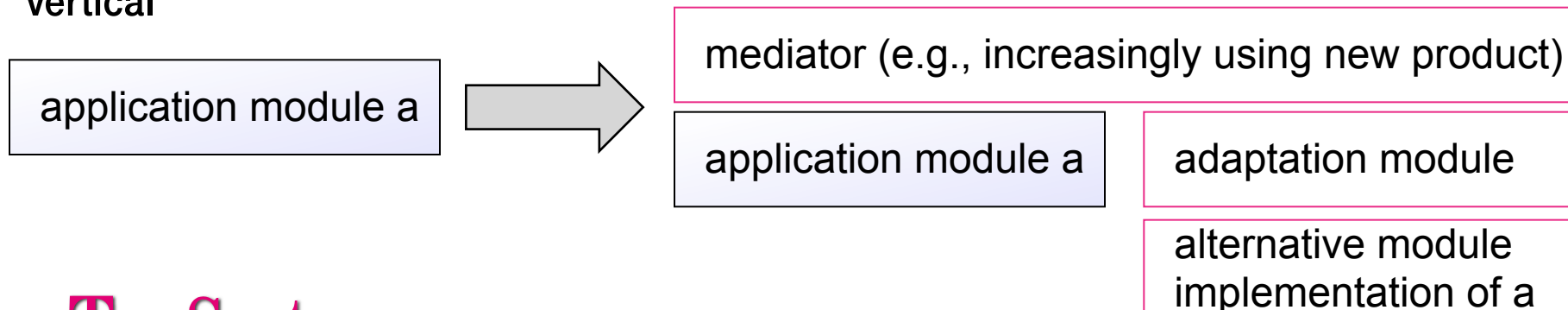
# CCMS Architecture as a Basis for Cloud-optimized CMSs. Elasticity.

- **Elasticity:** through dynamic setup changes, possible because of
  - uniform module interface (arbitrary module combinations),
  - Separation of Concerns (isolated base functionalities), and
  - stateless modules (dynamic increase / decrease of the number of redundant modules).

## horizontal



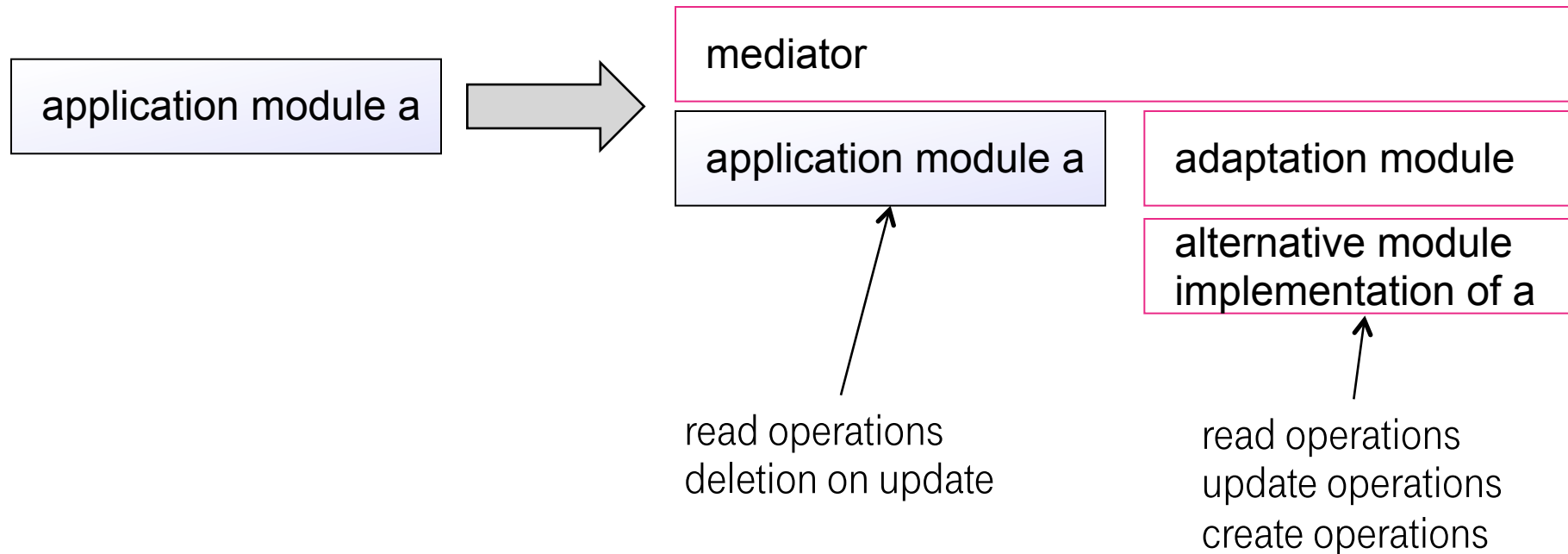
## vertical



# CCMS Architecture as a Basis for Cloud-optimized CMSs.

## Risk of vendor lock-in.

- **Risk of vendor lock-in:** leveraged for both local middleware and Cloud environment, depending on generators.
- Implemented the same way as vertical scaling by using a different implementation, but can accept slower pace of content migration.



# Summary and Future Work.

# Summary.

- Moving components of an overall systems to the Cloud raises severe security **problems**.
- Until **secure Intercloud** solutions become available, individual measures have to be taken.
- **Cloud-optimized software** should provide the necessary means.
- The development of Cloud-optimized software should, at least in the case of CMSs, be based on **model-driven development (MDD)** tools.
  - A central, coherent model.
  - Code generators that ensure the establishment of security measures.
  - Code generators that allow software evolution while maintaining those measures.
  - A target architecture that allows plugging in orthogonal services.
- The MDD approach developed for **Concept-oriented Content Management (CCMS)** systems fulfills these requirements.
- In any case, relevant parts of an existing overall architecture need to be rethought.



## Future Work.

- T-Systems MMS developed a **Cloud-CMS solution** based on the CoreMedia CMS. First customers use this solution.
- Currently there are efforts to **port** this solution to other CMS products.
- Practical insights are yet to be gained.
  
- First steps towards Intercloud solutions are made. This **market** needs to be monitored in order to employ or integrate off-the-shelves components where beneficial.
  
- The application of the CCMS approach for Cloud applications is not in productive use. Further **experiments** are required.
- A second step would be the development of adequate **management tools**.



Thank you.



**Dr. Hans-Werner Sehring**  
Architect Principal

**T-Systems Multimedia Solutions GmbH**

Hausanschrift	Binderstraße 26, D-20146 Hamburg
Telekontakte	Telefon +49 (40) 570062 8049
	E-Mail hans-werner.sehring@t-systems.com

Linked  <http://www.linkedin.com/in/hwsehring>

 [https://www.xing.com/profile/HansWerner\\_Sehring](https://www.xing.com/profile/HansWerner_Sehring)

