Selecting Multimedia Service Compositions in Mobile Environments

Wolf-Tilo Balke* and *Jörg Diederich
L3S Research Center, Germany

ICWS'06
Chicago, IL, USA

Outline

- Motivation
- Multimedia Service Composition
  - Mobile Constraints
- Quality- and Cost-based Selection Model
- Use Case
- Summary and Future Work

Motivation

- Multimedia applications today are **monolithic**
  - Service-oriented framework introduce efficient reusability of components
  - Services introduce flexibility and best effort approaches
  - Service routing through network from media server to end user

- Problem in Multimedia Services
  - How to organize highly complex and dynamic workflows?
  - How to exploit the deep understanding of multimedia data?

  - ***Web community* understands services –**
    ***Multimedia community* understands data**

Motivation

- **Limited capabilities** in end user devices
  - Computationally complex or power-demanding tasks have to be moved to powerful servers
  - But also environment changes: movement of device,...

- Basic Idea: **E²Mon Algorithm**
  - Monitors the execution chain of Web services
    - Graceful recovery from individual service failures
    - Graceful recovery from network-/device-specific alarms
  - Dynamically chooses the quality- and cost-optimal composition
    - Successive and parallel execution

Multimedia Service Composition

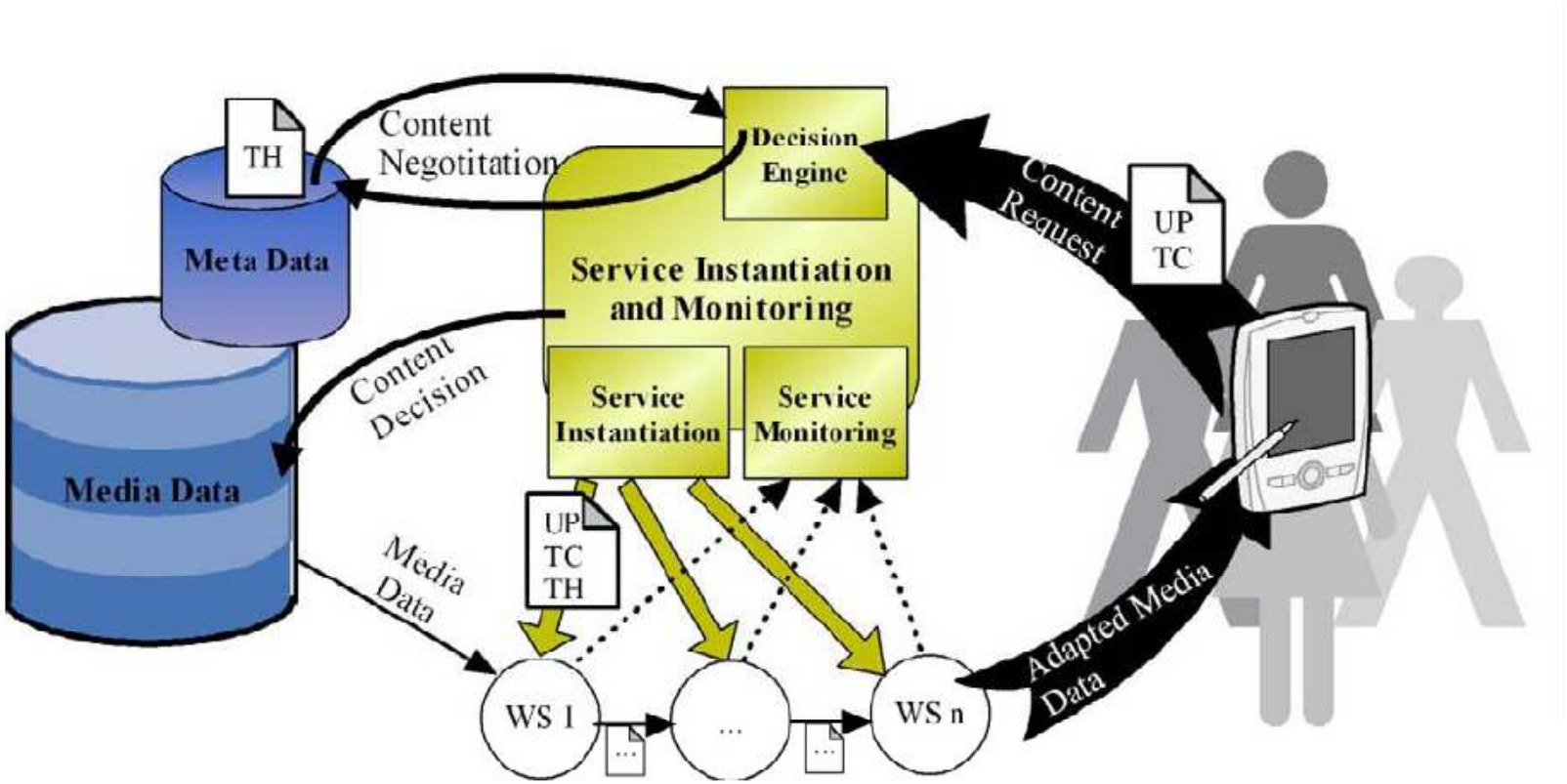
- Multimedia service composition is a composition **process**
  - Multiple services (e.g., retrieval, transcoding, display services) for processing and communication of multimedia data
  - Connected via functional and data dependencies to create a new multimedia service (e.g., a video-on-demand service)
  - Span over heterogeneous network and distributed system infrastructures
- Multimedia applications are usually **flow-based**
  - Data mostly continuous streams (e.g., video and audio streams)
  - Dependent in time and space
  - Stringent timing and spatial constraints on the functional services
  - Quality constraints (non-functional parameters) need to be taken into account

Multimedia Service Composition

- A **multimedia service** is a functional entity
  - With time, space and dependency relations to other services that precede or follow the application service
  - Functional dependency relations among individual multimedia services form a **service graph**
  - Service descriptions are expressed via meta-data and published in order for other services to be discovered and used
- **Mobile constraints**
  - Plethora of devices with different needs: data formats, …
  - Very limited capabilities: display, computing power, …
  - Power management: energy saving, battery low, …
  - Transport difficult: low bandwith, high costs, …

Multimedia Service Composition

- Controlled by central instantiation and monitoring service



Some assumptions

- **No malicious services**
  - Capabilities are described correctly (MM-specific)
  - Provider discloses correct costs, QoS parameters, etc.
- Service implementations for specific Web service types can be used **interchangeably**
  - Data is understood according to MPEG-7/21 descriptions
- **Central** control instance (monitor/proxy) caching results of previously executed Web services
  - Reuse in failure recovery

E²Mon Algorithm

- Run as a Web service on client or proxy machine
  - Proxy reduces communication load of mobile device for discovery, invocations, etc.
  - Efficient if client alarms are rare compared to external events (service failures,…)
- **4 basic phases**
  - Workflow enumeration
  - Service discovery
  - Service chain selection
  - Execution monitoring

E²Mon Algorithm

- **Workflow enumeration**
  - Construct all possible workflows anticipated for multimedia applications (usage patterns)
  - **Input, output and intermediate/transformational services**
  - Languages for workflow executions: BPEL4WS, …
  - Languages for service capabilities: OWL-S, WSDL-S, …
- More complex enumeration
  - AI planning techniques
  - State machines or Petri net-based approaches for simulation and verification

E²Mon Algorithm

- **Service discovery**
  - Service descriptions with multimedia metadata
  - Discover all suitable implementations
  - Classify due to constraints met/compromises needed
  - Extended UDDI, cooperative discovery
- Periodic rediscovery
  - Rediscovery on failure is too time-consuming (prefetching)
  - Execution parameters can change (e.g., QoS) more cost-effective solutions should be instantiated (hand-over)
  - Less important alarms (battery-low, etc.) can be handled, if better chains with respect to the alarm are available

E²Mon Algorithm

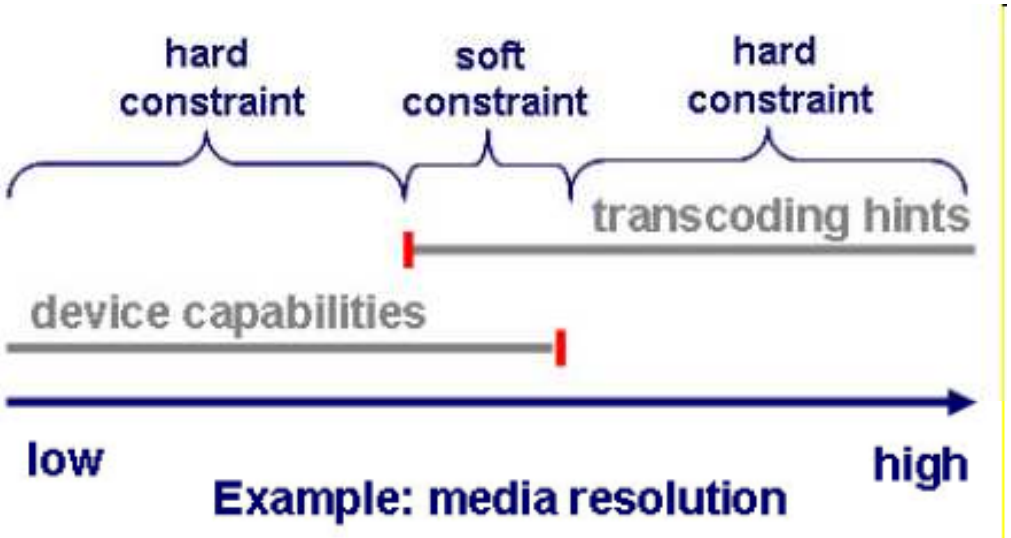
- **Service chain selection**
  - Calculate costs for all service chains
  - **Quantitative:**

    e.g., $F(I,t) = M(I) + Q(I) + w(t) * P(I) + U(I)$
  - $M(I)$ – typical provider costs (service use, network use, etc.)
  - $Q(I)$ – quality of service costs (resolution, bandwidth, etc.)
  - $P(I)$ – power consumption costs, weighted by time variant function $w(t)$ dependent on device
  - $U(I)$ – user preference-based costs (decrease in MPEG-7/21 preference value, etc.)

- **Problem: very complex and sensitive functions**

E²Mon Algorithm

- **Preference-based** service chain selection
  - Get metadata given by MPEG-7/21 part 7 usage environment for digital item adaptation (DIA)
  - Metadata can include *transcoding hints*, *user interaction tools*, and *terminal capabilities*
- Advanced digital item adaptation
  - Handle complex preference trade-offs **qualitatively**
  - Still problematic in terms of efficiency

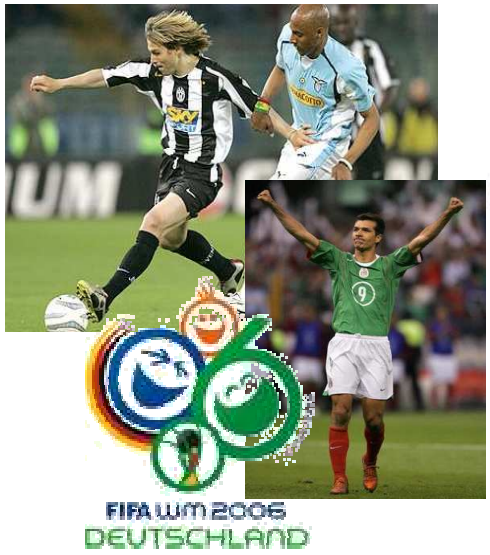


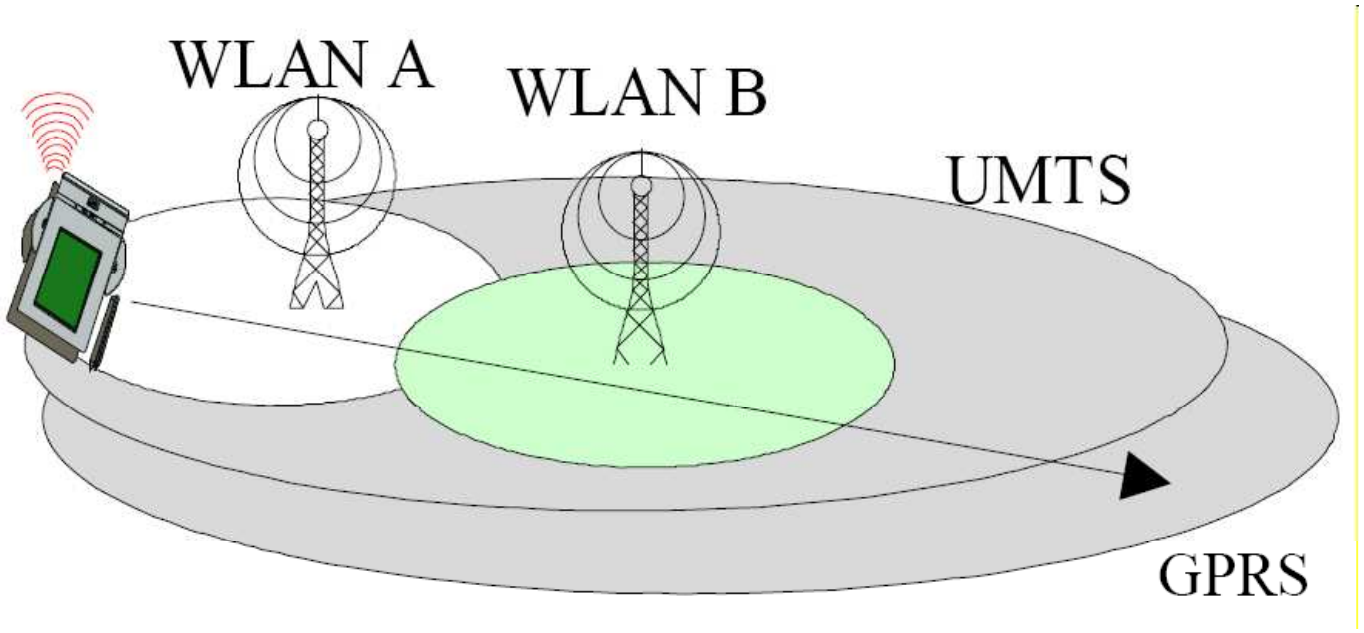
E²Mon Algorithm

- **Execution monitoring**
  - Execute services in a supervised fashion
  - Keep alternative chains updated (periodic rediscovery)
  - Handle service failures by reassessing cost for the same chain with the next best implementation of the failed service
  - Change to different implementation or different chain
- Possible alarms
  - Service failure needs graceful recovery
  - Change in service parameters, especially QoS
  - Another more cost-effective chain becomes executable
  - Local events (roaming to new network, battery warning, user interaction, ...)

Media Streaming Use Case

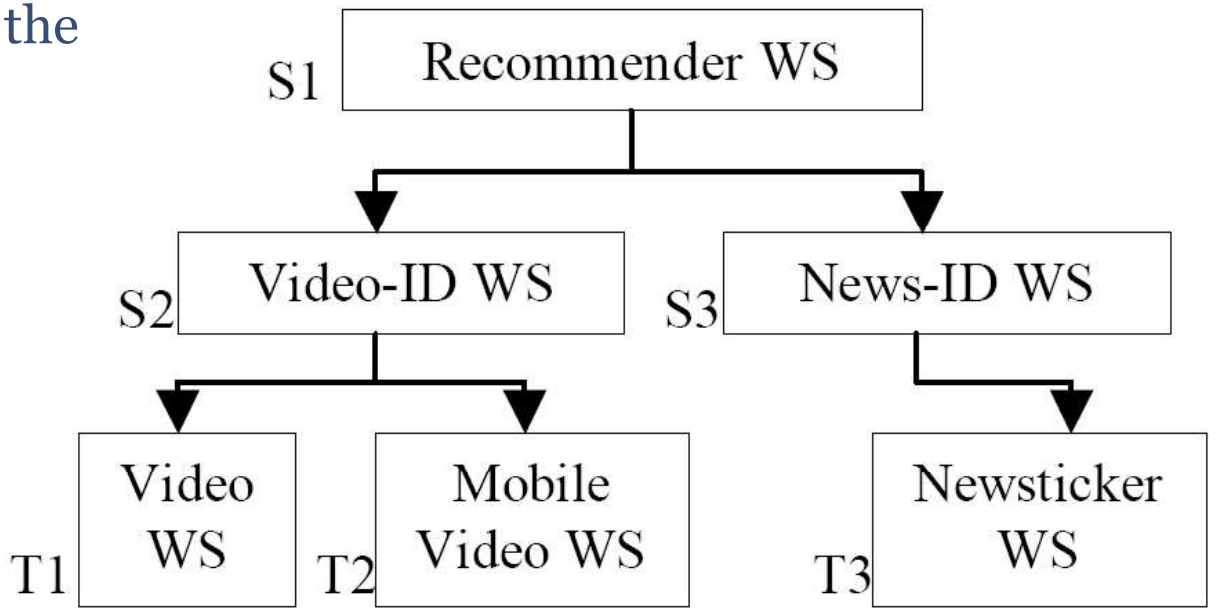
- FIFA Soccer WorldCup 2006
  - All games streamed using digital video broadcasting standard (DVB-T / DVB-H)
  - Moving through networks with a mobile device (PDA)
  - Small screen, network technologies: WLAN, UMTS, GPRS





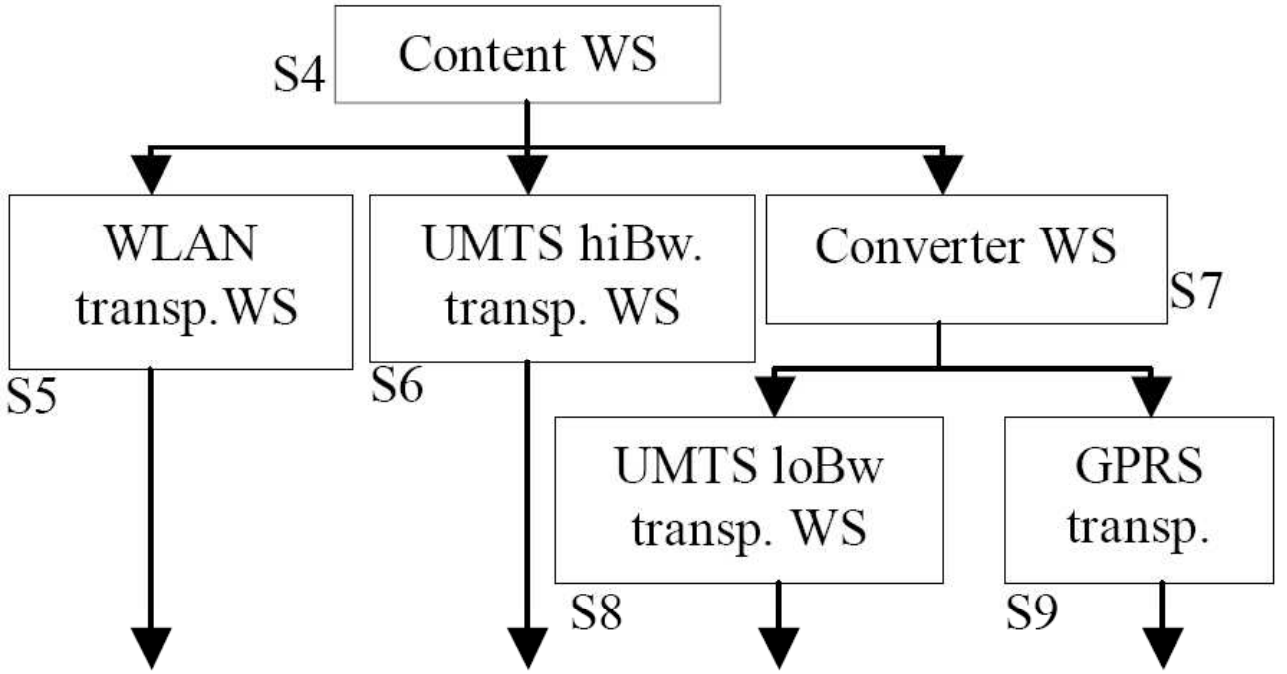
Media Streaming Use Case

- Enumeration of possible service chains
  - Planning is not the major problem of multimedia applications
  - Mixture of **content** and **transport** services
  - Hard constraints might **rule out** some of the compositions



Media Streaming Use Case

- Sequential and parallel execution, e.g. video transportation service T1
  - Bundles different transportation technologies, transcoding, etc.
  - But also: billing services, etc.



Media Streaming Use Case

- **Possible service chains**
  - Service descriptions enable determining executable sequences
  - Different implementations/ providers may exist
  - Services can interpret data by exploiting meta-data (MPEG-7/21)

Nr.	Service chain	Description
1	S1-S2-S4-S5	Video-WLAN
2	S1-S2-S4-S6	Video-UMTS-hiQ
3	S1-S2-S4-S7-S8	Video-UMTS-lowBw
4	S1-S2-S4-S7-S9	Video-GPRS-lowBw
5	S1-S2-S10-S9	Mobile Video-GPRS
6	S1-S3-S11-S5	Newsticker-WLAN
7	S1-S3-S11-S9	Newsticker-GPRS

Media Streaming Use Case

- Compare possible chains and implementations

Nr.	1	2	3	4	5	6	S_{exec}
1	S1	S2	S4	S5	nil		True
2	S1	S2	S4	S6	nil		True
3	S1	S2	S4	S7	S8	nil	True
4	S1	S2	S4	S7	S9	nil	True
5	S1	S2	S10	S9	nil		**False**
6	S1	S3	S11	S5	nil		True
7	S1	S3	S11	S9	nil		True

Executebility information

Service chains

Service index

S1	S2	S4	S5	S6	S7	S8	S9	S3	S11
I1	I2	I4a	I5a	I6	I7a	I8	I9	I3	I11
		I4b	I5b		I7b				

Media Streaming Use Case

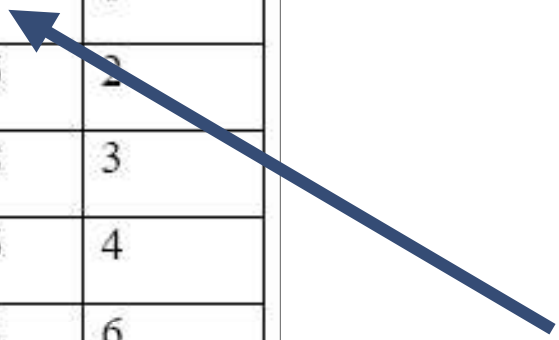
- Cost-table
  (w/o preference values)
- Based on implementations

Implementation	M	Q	P	U	**F(I, t) w(t) = 0**
I1	0	0	2		**0**
I2	1	0	2		**1**
I4a	0	0	64	4	**4**
I4b	1	0	64	1	**2**
I5a	0	0	4		**4**
I5b	1	0	4		**5**
I6	15	1	8		**23**
I7a	2	4	0		**6**
I7b	3	2	0		**5**
I8	6	4	4		**10**
I9	3	15	4		**18**
I3	1	0	2		**1**
I11	1	17	2	0	**18**

Media Streaming Use Case

- Calculate $F(I, t)$ for all chains

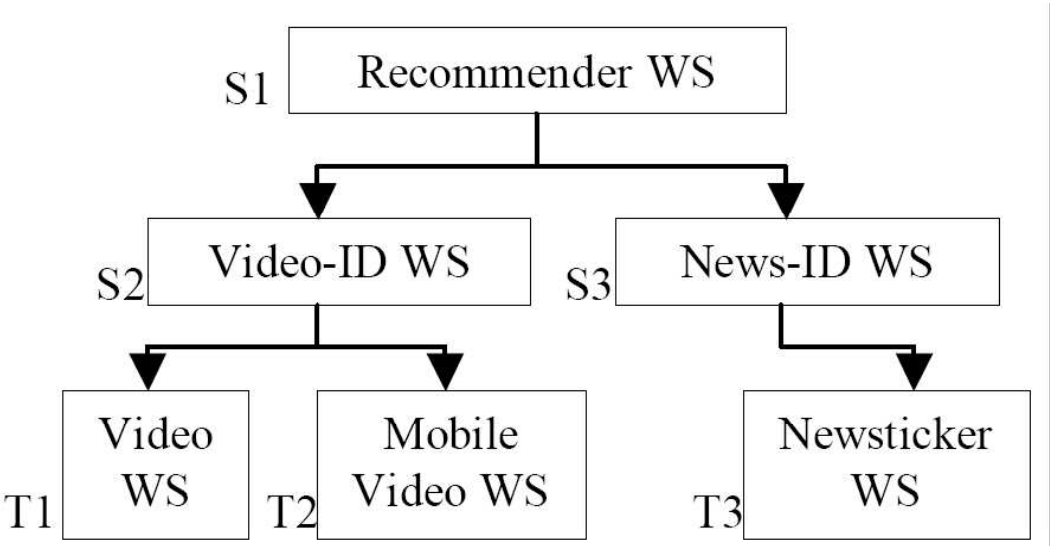
Nr.	1	2	3	4	5	6	V_{min}	V_{chain}
1	I1	I2	I4b	I5a	nil		7	1
2	I1	I2	I4b	I6	nil		26	2
3	I1	I2	I4b	I7b	I8	nil	18	3
4	I1	I2	I4b	I7b	I9	nil	26	4
5	I1	I3	I11	I5a	nil		23	6
6	I1	I3	I11	I9	nil		37	7



Minimal cost chain is instantiated

Media Streaming Use Case

- Monitoring dynamics
  - Service chains change executebility status, e.g., joining or leaving a network (WLAN, GPRS)
  - Discovery of new service implementations, e.g., mobile video service
  - Local events, e.g. low battery changes best possible service from streaming to newsticker (preference vs. energy)



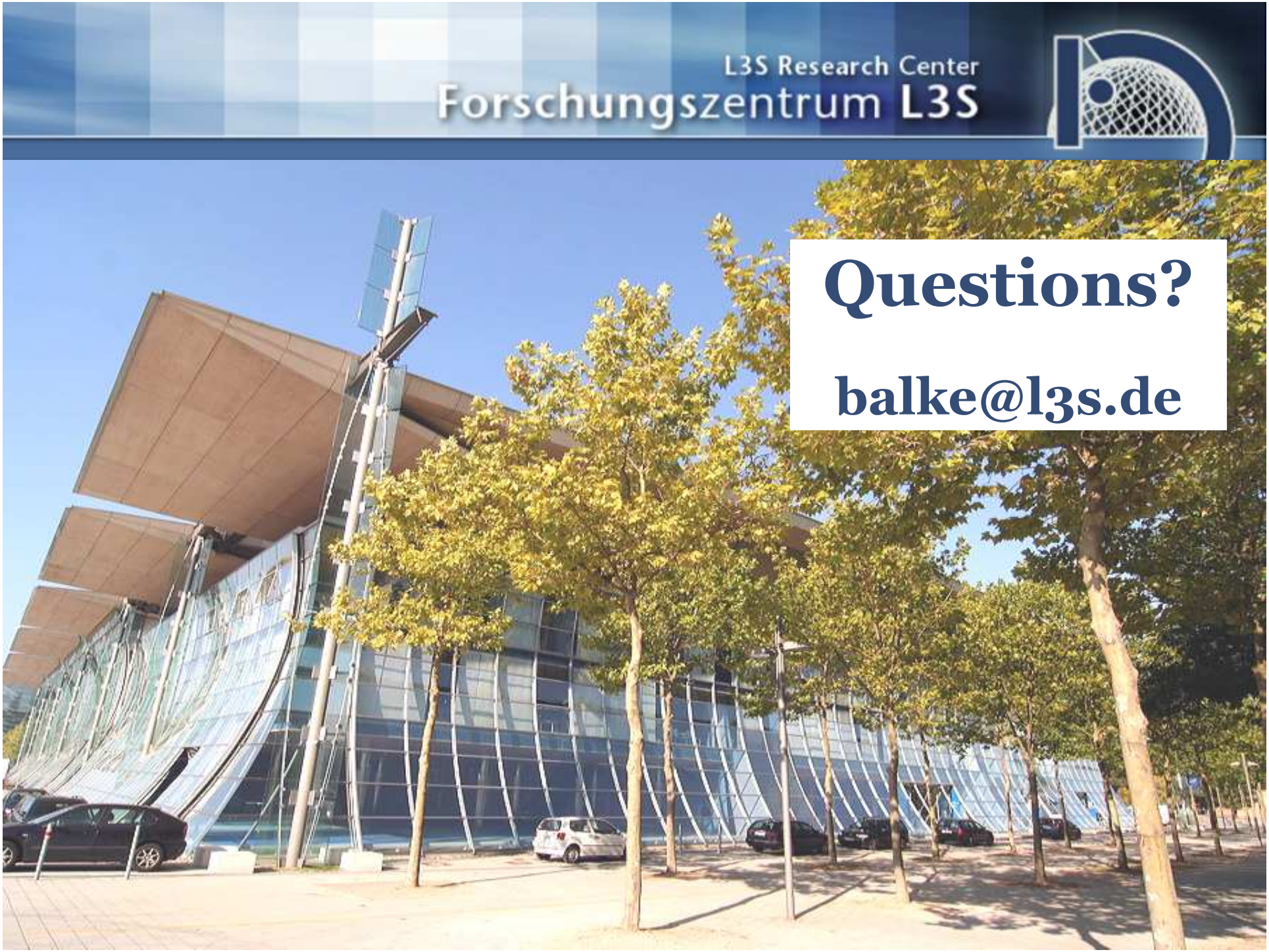
	I1	I2	I4a	I4b	I7a	I7b	I9	I3	I11	I10
$F(I,t)$, $w(t)$=0	0	1	4	2	6	5	18	1	18	5
$F(I,t)$, $w(t)$=1	2	3	68	65	6	5	22	3	20	35

Summary

- Problem
  - Multimedia Services demand deep understanding of data and flow-based compositions
  - Mobile devices have limited capabilities and pose additional constraints
  - Environmental parameters may change quickly
- Idea
  - E²Mon algorithm selects the most cost-efficient chain for instantiation
  - Continuously monitors the execution and handles alarms
  - Dynamically controls costs based on providers, quality of service, power consumption and user preferences

Future Work

- Further investigation of E²Mon's applicability
  - Currently a **large testbed** is built to investigate scalability constraints
  - Find good application-dependent **defaults** for cost functions and parameters (e.g., optimal rediscovery time)
  - Consider **qualitative** functions for service selection and introduce new more expressive preference model


L3S Research Center
Forschungszentrum L3S
Questions?
balke@l3s.de