Event- and Notification based Interaction

Middleware for IPC in DS

Client-Sever Relation is the most common form of IPC

RPC e.g. for remote file access
CORBA and Java RMI

Peer-to-Peer Relation is the (next?) big step towards more scalable systems

Event and Notification Services

Event and Notification Services

What is an event?

How do I know about an event?

How to define constraints on event distribution?

How are events structured?

Event and Notification Services

topology affects scalability

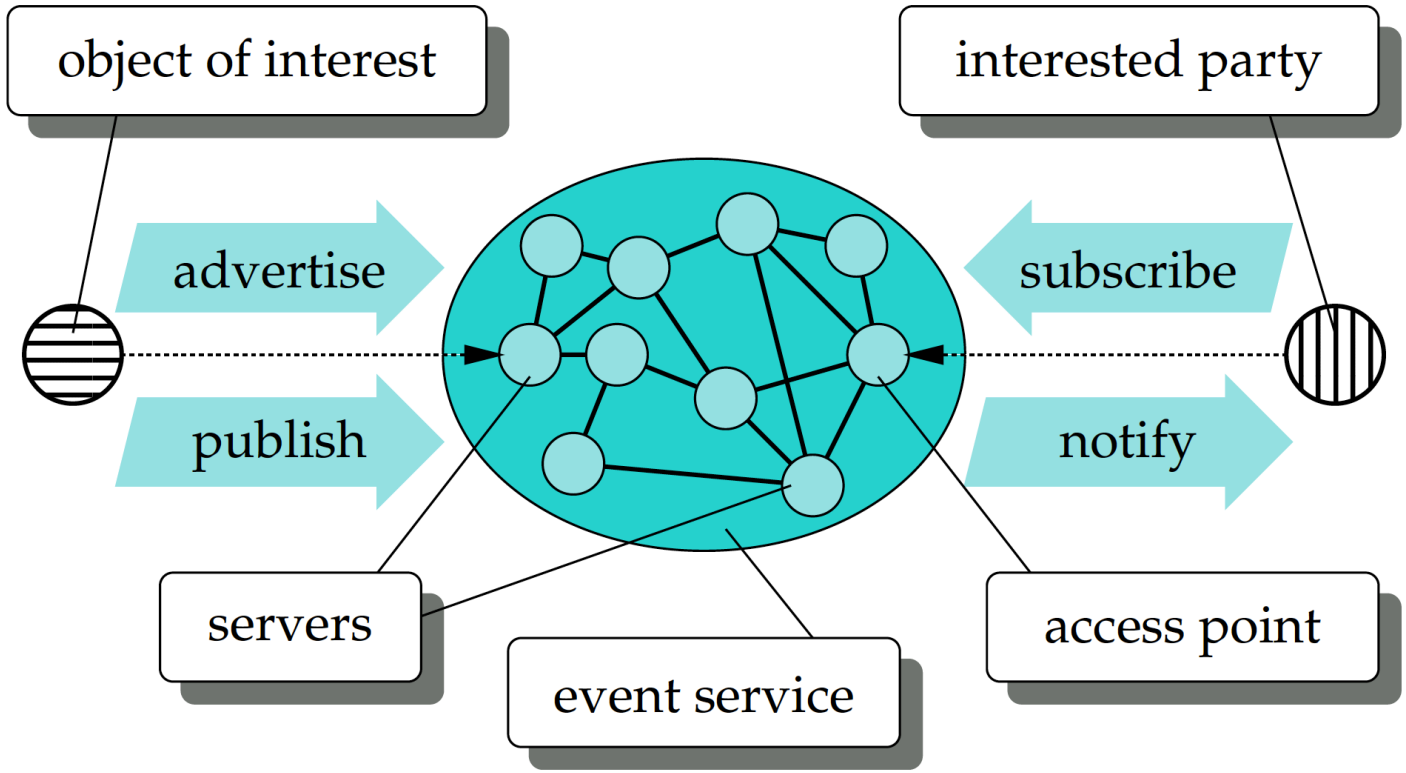
subscription specs affect expressiveness

Subscription Language	Architecture: *centralized*	Architecture: *hierarchical client/server*	Architecture: *general peer-to-peer*
channel-based	Field [15] CORBA Event Service [13] Java Dist. Event Spec. [19]	CORBA Event Service [13]	IP multicast [6] iBus [18]
subject-based	ToolTalk [9]	NNTP [10] JEDI [5] TIB/Rendezvous [20]	*(none)*
content-based	Elvin [17]	Keryx [21] Yu et al. [22]	Gryphon [1]
content-based with patterns	GEM [12] Yeast [11] CORBA Notification Service [14] object-oriented active databases [4]	SIENA	SIENA

Classification of Event-Based Technologies

from: Antonio Carzaniga, David S. Rosenblum, Alexander L.Wolf: Achieving Scalability and Expressiveness in an Internet-Scale Event Notifivation Service

The SIENA Distributed Event Notification Service



from: Antonio Carzaniga, David S. Rosenblum, Alexander L.Wolf: Achieving Scalability and Expressiveness in an Internet-Scale Event Notifivation Service

Example of Event Notification and Filtering

string	*class*	= *finance/exchanges/stock*
time	*date*	= *Mar 4 11:43:37 MST 1998*
string	*exchange*	= *NYSE*
string	*symbol*	= *DIS*
float	*prior*	= *105.25*
float	*change*	= *-4*
float	*earn*	= *2.04*

Notification

string	*class*	> ∗ *finance/exchanges/*
string	*exchange*	= *NYSE*
string	*symbol*	= *DIS*
float	*change*	< *0*

a Filter

Filtering and Attribute Matching in SIENA

An **attribute** α _ ($type_\alpha$, $name_\alpha$, $value_\alpha$)
matches an **attribute constraint**
Φ = ($type_\Phi$, $name_\Phi$, $operator_\Phi$, $value_\Phi$)
iff
$type_\alpha = type_\Phi \wedge name_\alpha = name_\Phi \wedge operator_\Phi (value_\alpha, value_\Phi)$

We say an attribute α of a notification *n* matches a constraint Φ expressed by a filter f and write in short: $\phi \sqsubset_f^n \alpha$

Thus a notification n matches a filter f if:

$$f \sqsubset_S^N n \Leftrightarrow \forall \phi \in f : \exists \alpha \in n : \phi \sqsubset_f^n \alpha$$

Filtering and Attribute Matching in SIENA

Examples of notification filtering

The relation $\sqsubset_A^N$ defining the set of notifications by a subscrption:

$$f \sqsubset_S^N n \Leftrightarrow \forall \phi \in f : \exists \alpha \in n : \phi \sqsubset_f^n \alpha$$

subscription		*notification*
string what = alarm	$\sqsubset_S^N$	*string* what = alarm *time* date = 02:40:03
string what = alarm *integer* level > 3	$\not\sqsubset_S^N$	*string* what = alarm *time* date = 02:40:03
string what = alarm *integer* level > 3 *integer* level < 7	$\not\sqsubset_S^N$	*string* what = alarm *integer* level = 10
string what = alarm *integer* level > 3 *integer* level < 7	$\sqsubset_S^N$	*string* what = alarm *integer* level = 5

Filtering and Attribute Matching in SIENA

advertisement		*notification*
string what = alarm *string* what = login *string* user *any*	$\sqsubset_A^N$	*string* what = alarm
string what = alarm *string* what = login *string* user *any*	$\not\sqsubset_A^N$	*string* what = alarm *time* date = 02:40:03
string what = alarm *string* what = login *string* user *any*	$\sqsubset_A^N$	*string* what = login *string* user = carzanig
string what = alarm *string* what = login *string* user *any*	$\not\sqsubset_S^N$	*string* what = logout *string* user = carzanig

Examples of advertisement filtering

The relation $\sqsubset_A^N$ defining the set of notifications by an advertisement:

$$a \sqsubset_A^N n \Leftrightarrow \forall \alpha_n \in n : \exists \phi_a \in a : \phi_a \sqsubset_f^n \alpha_n$$

Event Patterns and Sequence

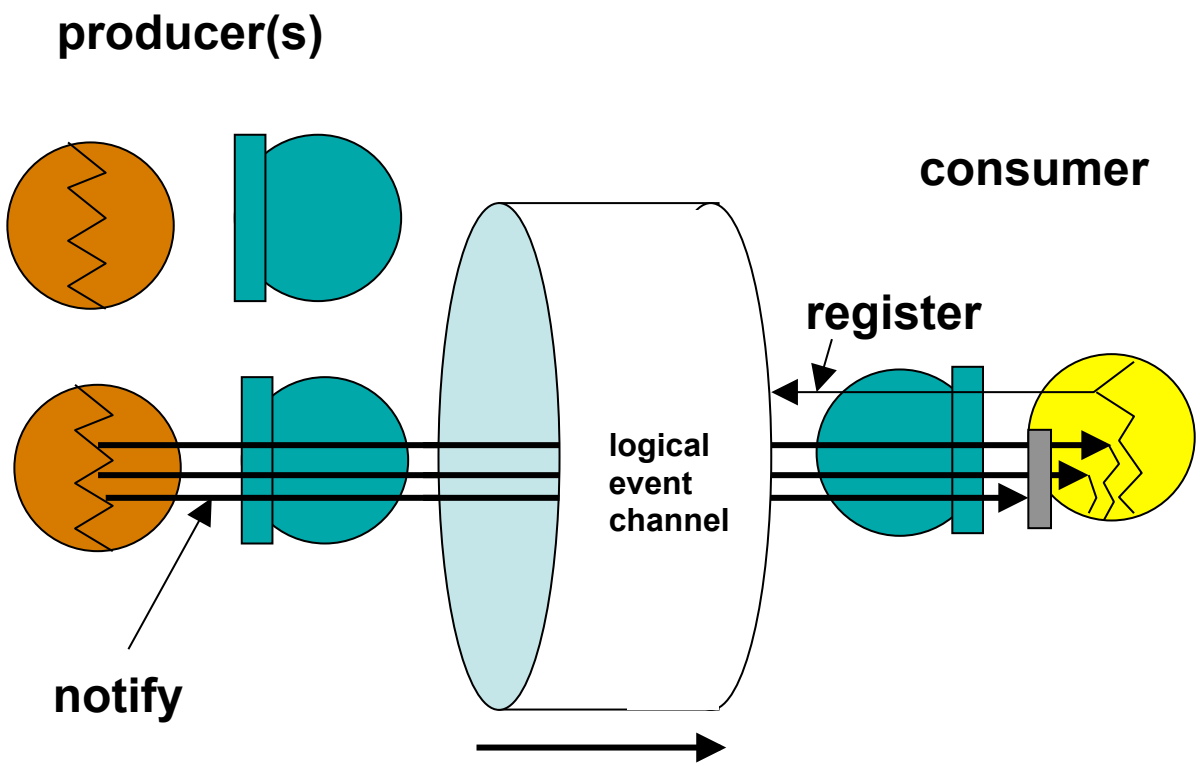
$string \quad what > * finance/exchanges/$
$string\ symbol = MSFT$
$float \quad change > 0$

.

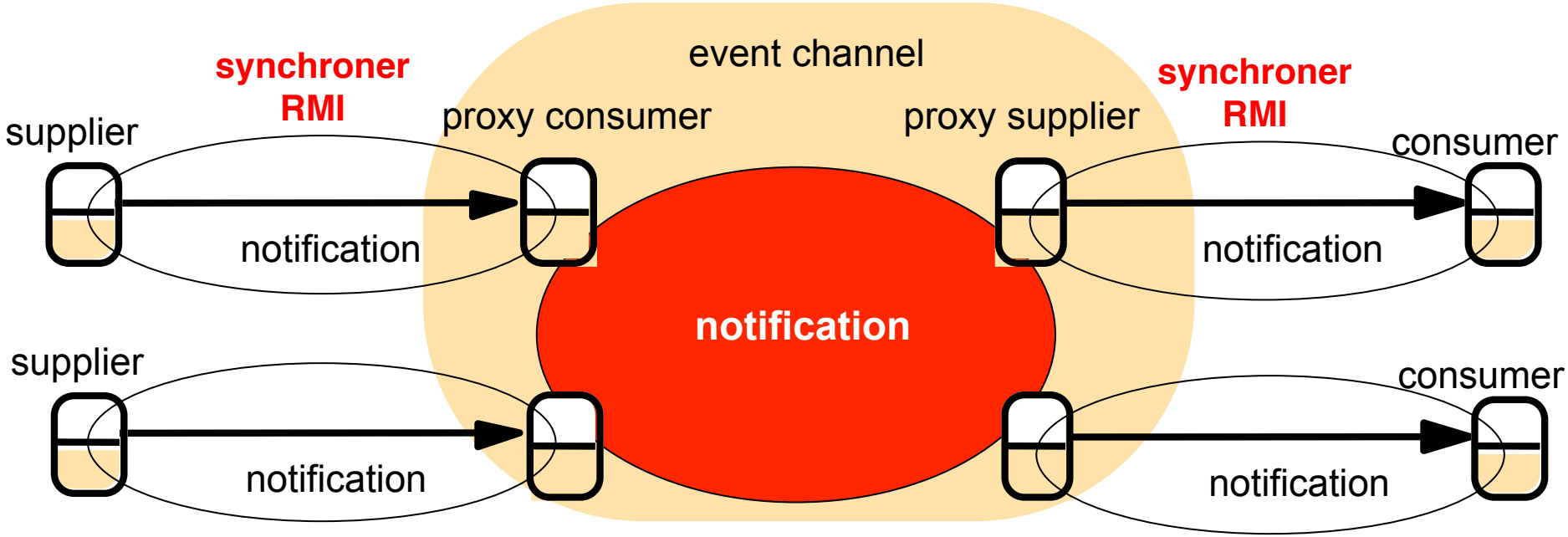
$string \quad what > * finance/exchanges/$
$string\ symbol = NSCP$
$float \quad change > 0$

time

Corba Event Service



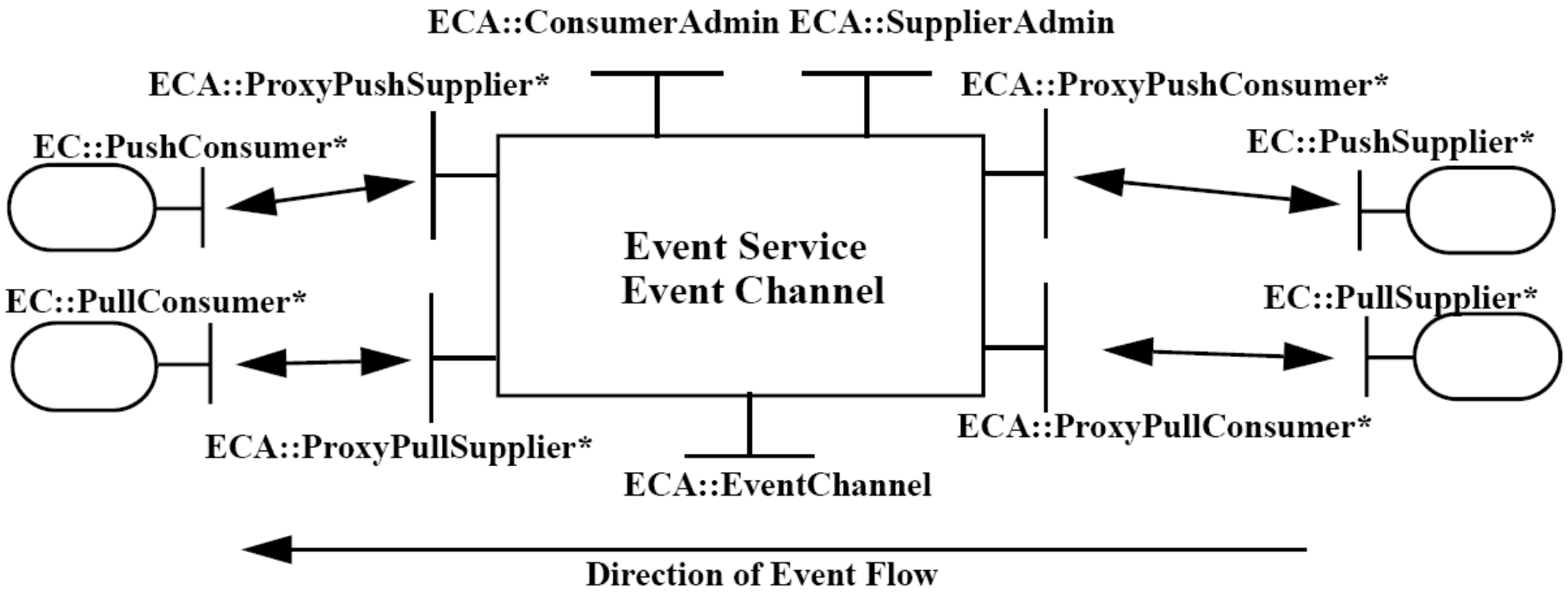
Corba Event Service



Asynchronous, anonymous event transfer from the supplier to the consumer.

push supplier, push consumer: supplier originated event transfer.
pull supplier, pull consumer: consumer originated event transfer.

Corba Event Service



Corba Event Service

Limitations of the event channel:

1. **supports no event filtering capability, and**
2. **no ability to be configured to support different qualities of service.**

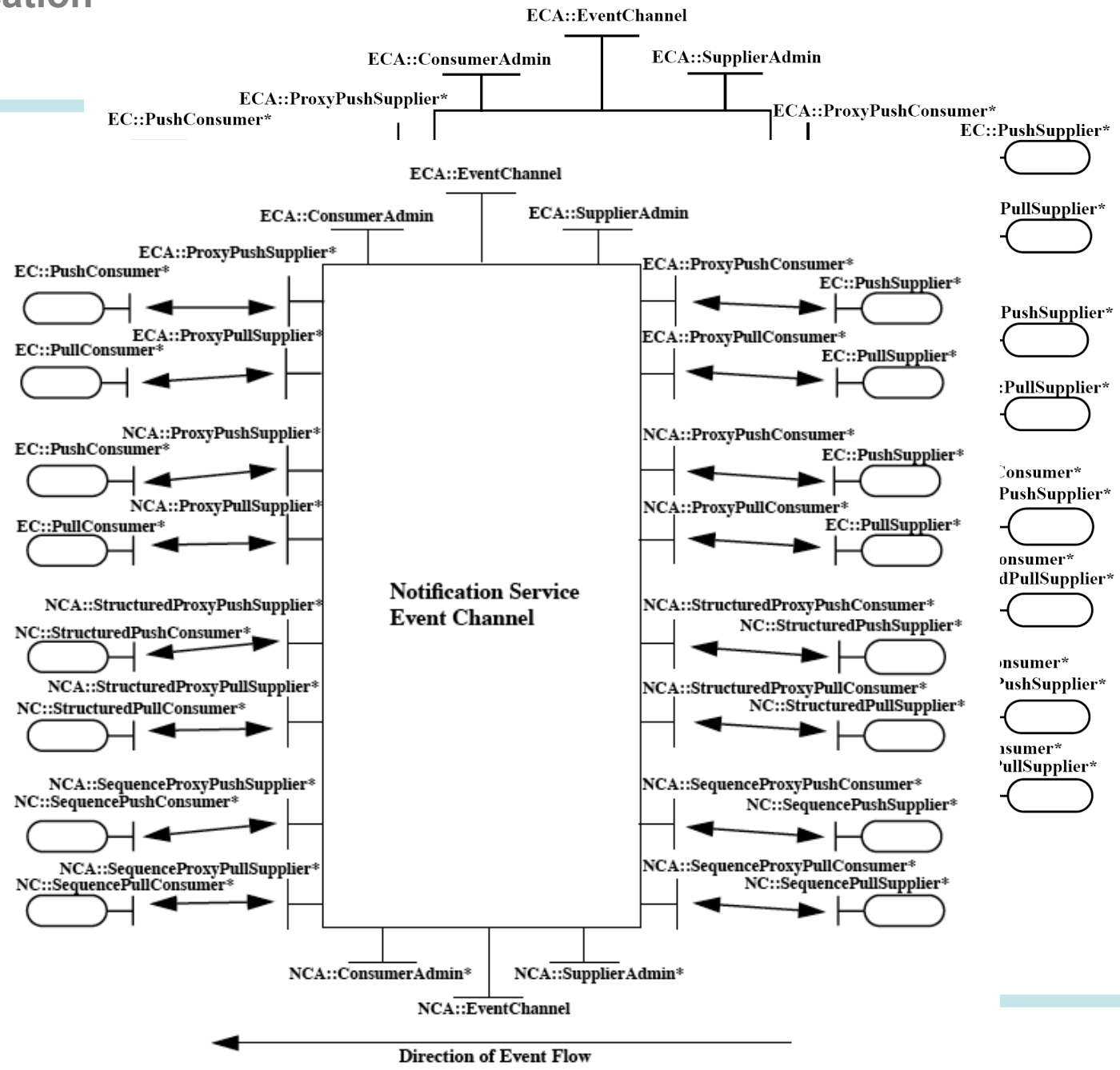
The Notification Service enhances the Event Service by introducing the concepts of filtering, and configurability according to various quality of service requirements.

Corba Notification Service

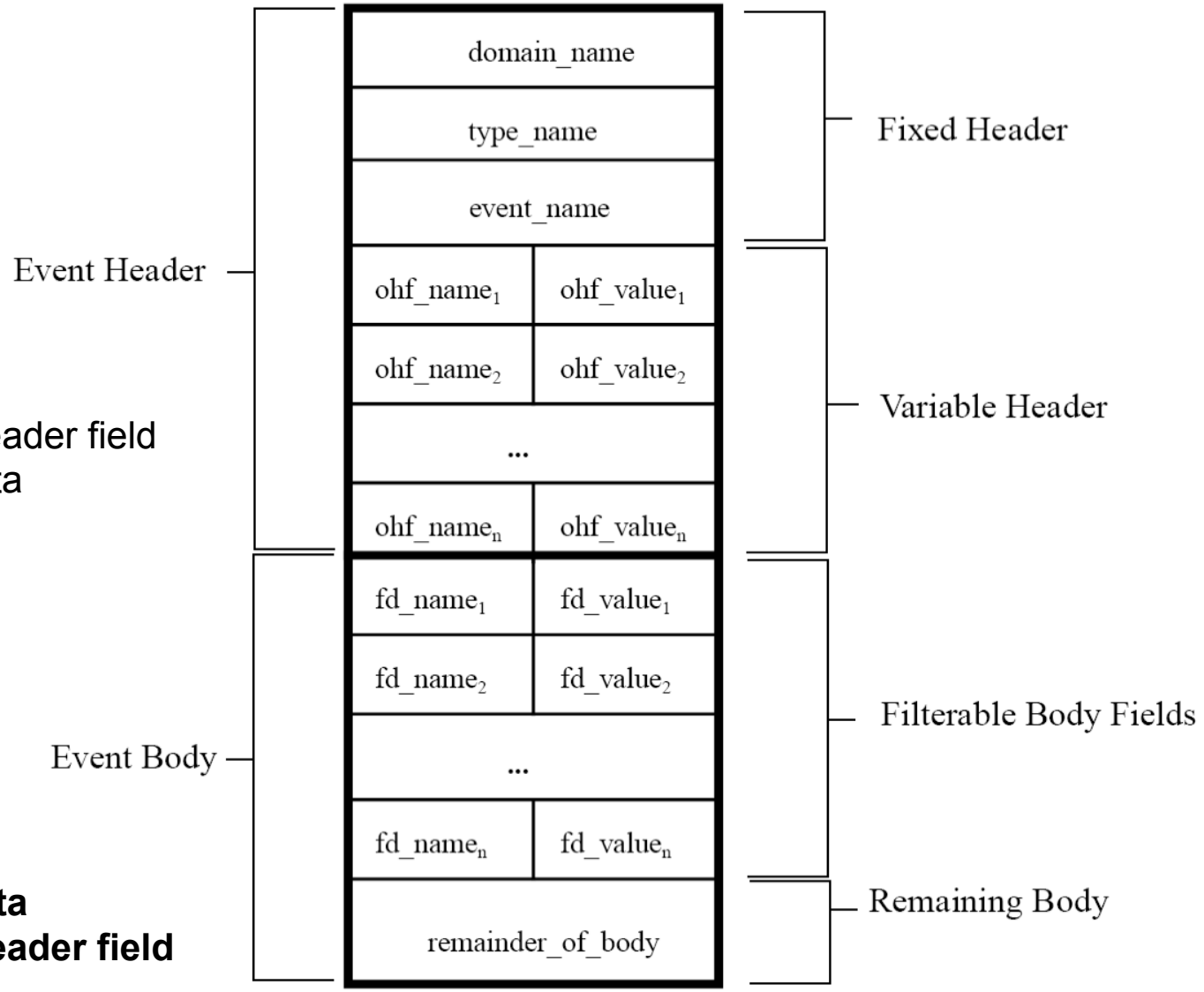
extends event service by:

- **consumers can define filter objects to define which events they are interested in.**
- **quality properties of a channel can be configured, e.g. reliability properties or order preferences like FIFO or priorities.**
- **consumer can detect event types which are advertised by producers.**
- **producers can discover interests of the consumers**
- **optional event-type repository allows access to event structures. Supports definition of filter contraints.**

Corba Notification Service



The structure of a Structured Event



ohf: optional header field
fd: filterable data

fd: filterable data
ohf: optional header field

Standard optional header fields

Header field name	Type of associated value	Meaning
EventReliability	short	This value portion of this header field has two well defined settings: 0 means "best effort"; 1 means "persistent". If set to 0, event can be treated as non-persistent and lost upon failure of the channel. At least one attempt must be made to transmit the event to each registered consumer, but in the case of a failure to send to any consumer, no further action need be taken. If set to 1, channel should make the event persistent, and attempt to retransmit upon channel recovery from failure. This setting only has meaning when ConnectionReliability is also set to 1, in which the combination essentially means guaranteed delivery.
Priority	short	Indicates the relative priority of the event compared to other events in the channel. Can take on any value between -32,767 and 32,767, with -32,767 being the lowest priority, 32,767 being the highest, and 0 being the default.
StartTime	TimeBase::UtcT	Gives an absolute time (e.g., 12/12/99 at 23:59) after which the channel can deliver the event.
StopTime	TimeBase::UtcT	Gives an absolute time (e.g., 12/12/99 at 23:59) when the channel should discard the event.
Timeout	TimeBase::TimeT	Gives a relative time (e.g., 10 minutes from time received) when the channel should discard the event). The special value *zero* indicates there is no timeout.

Corba Notification Service

Structured Event:

event header				event body	
domain name	type name	event name	optional h-field(s)	filterable body fields	rest

Example:

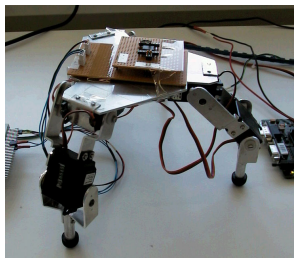
health care	patient supervision	heartbeat low alarm 22.06.06 9:30	priority 1o	ring alarm	blood pressure	respiration	rest

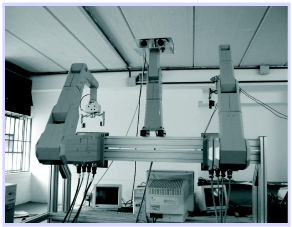
Embedded Systems

- Independence
- Resource Constraints
- Interaction with the environment
- Cooperation with other Systems

Desired Properties:

- Adaptivity
- Mobility
- Robustness and fault-tolerance
- Scalability and Extensibility





What Abstractions for Systems of Smart Devices ?

Smart Devices have resource constraints!

content-based would require analysis if every event
channel-based offers too little semantics
subject-based needs an efficient realization

Idea:

Exploit the inherent filtering properties of the communication hardware and bind subjects to message identifiers.

Events

Abstraction defining individual context and quality attributes

event:= <subject, funct_attr, non-funct_attr, value>

Functional Attributes are Environment Attributes and Context Related:

- time,
- location,
- operation mode,
- network zone, ...

Non-Functional Attributes are QoS related:

- deadline of dissemination
- validity interval
- criticality, ...

An Event Channel :

- Is identified by the content of information it disseminates
- Disseminates events of a certain subject
- Abstracts the underlying communication mechansim
- Is an identifyable object of the middleware at the API
- Is the object to assign QoS-attributes

Event Channels

Abstraction of the Infrastructure, i.e. explicit specification of the channel through which the events are disseminated

event_channel:= <subject, attribute_list, handlers>

attibute_list (related to the quality of the channel):

- latency attributes,
- reliability attributes,
- network zone, ...

handlers:

- notification handler
- exception handler

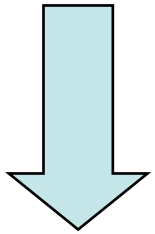
Description of an Event Channel

```
<EventChannel>
   <SubjectUID>0 xa4efeecc3210b497</SubjectUID>
      <Description>emergency switch</Description>
      . . .
      <Attributes>
      <Attribute>
            <Name>Omission</Name>
            <Value>2</Value>
      </Attribute>
      <Attribute>
            <Name>Period</Name>
            <Dimension>
                  <SIUni t><Seconds>1</Seconds></SIUni t>
                  <Magnitude>-3</Magnitude>
            </Dimension>
            <Value>50</Value>
      </Attribute>
  </Attributes>
. . .
</EventChannel>
```

Implementing the Publisher/Subscriber Model of Communication

***Elements of the model*:**

Publisher,
Subscriber,
Information instance (event)
Information type(subject)



Mapping the elements of the model to the elements of the underlying system

***Elements of the system*:**

Objects,
Messages,
Addresses

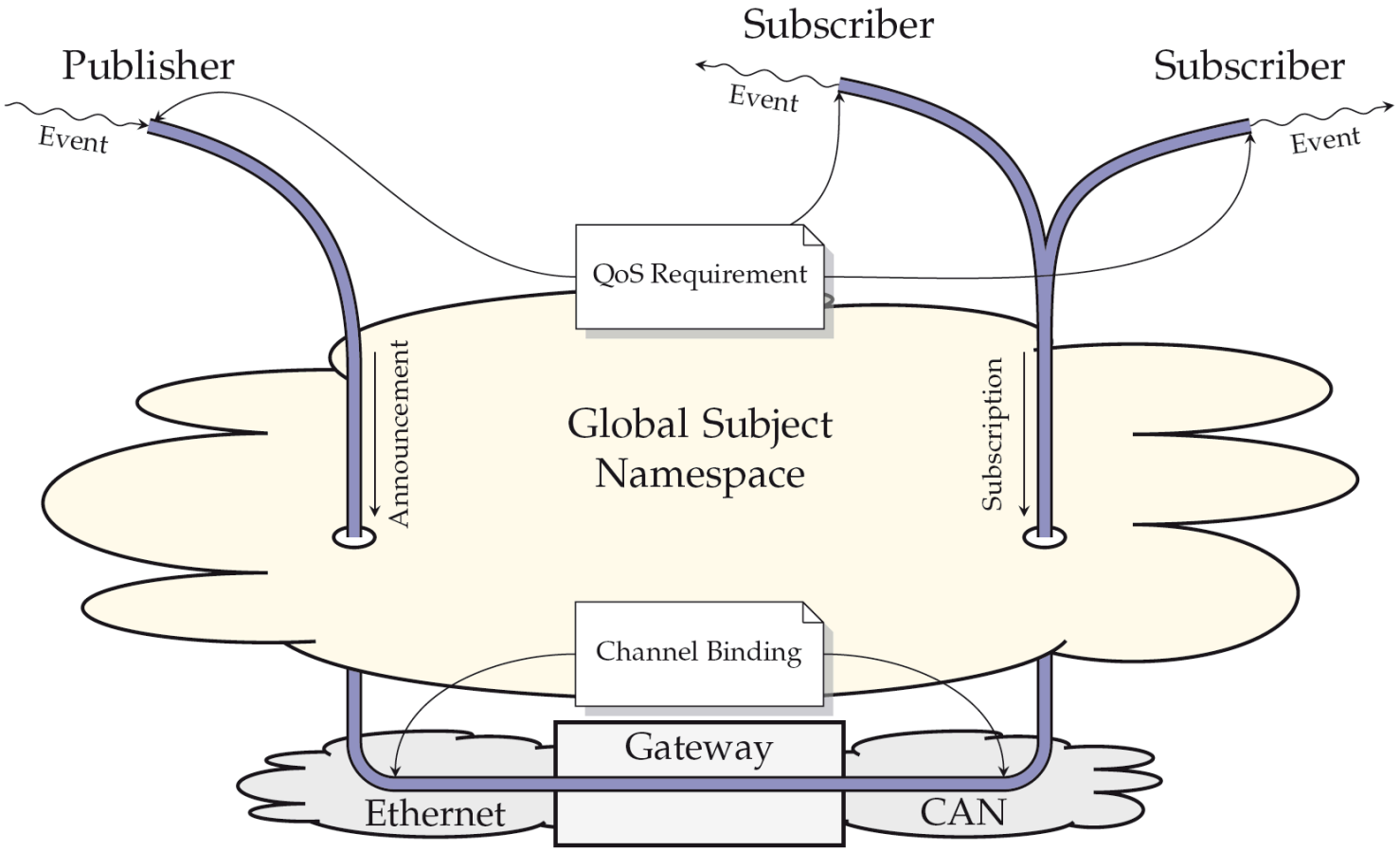
Implementation Issues

Routing: getting a message to its destination

Filtering: selecting the relevant messages on the receiver site

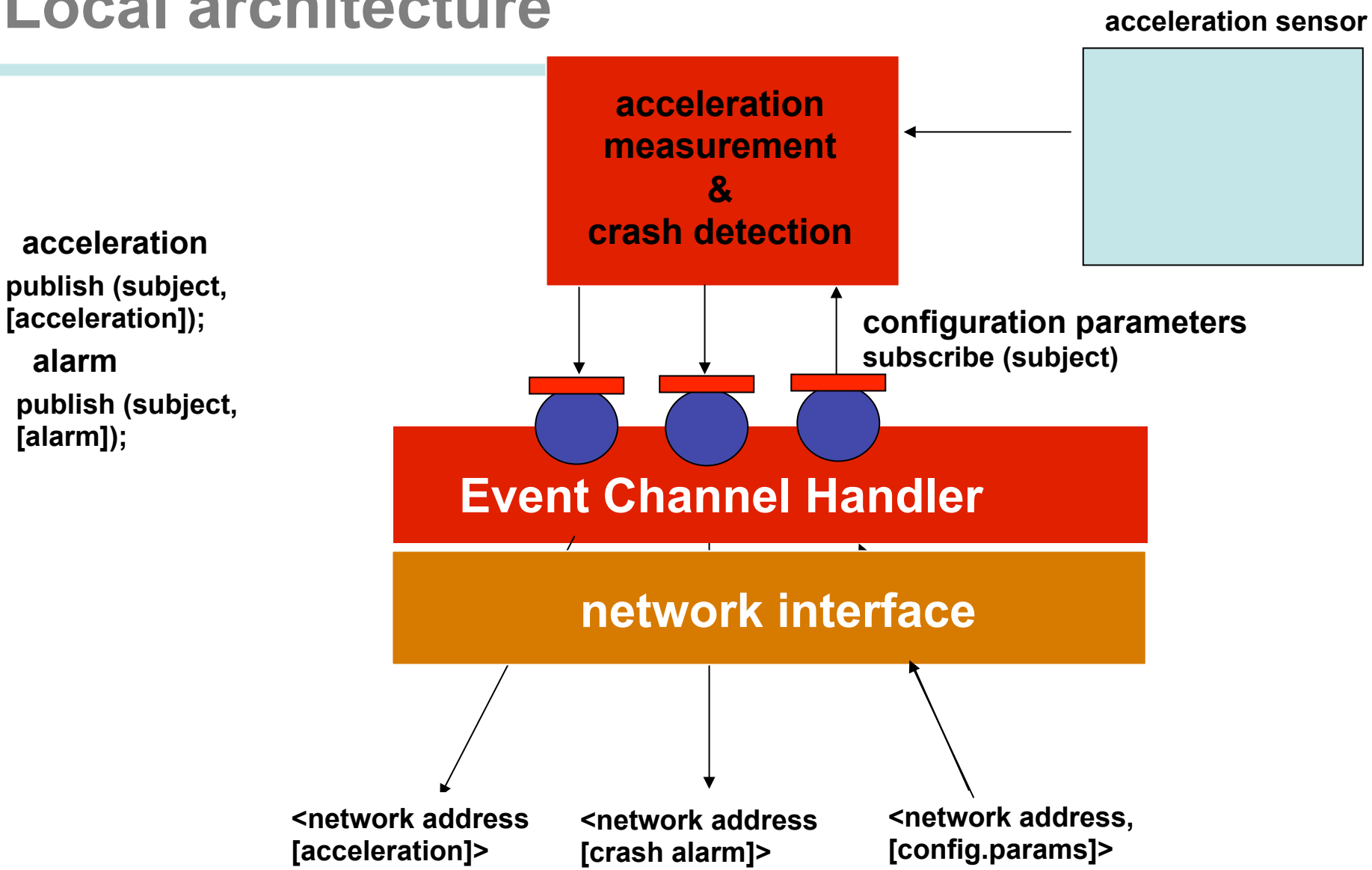
Binding: relating the content of a message to its address

The FAMOUSO Event-Based Middleware

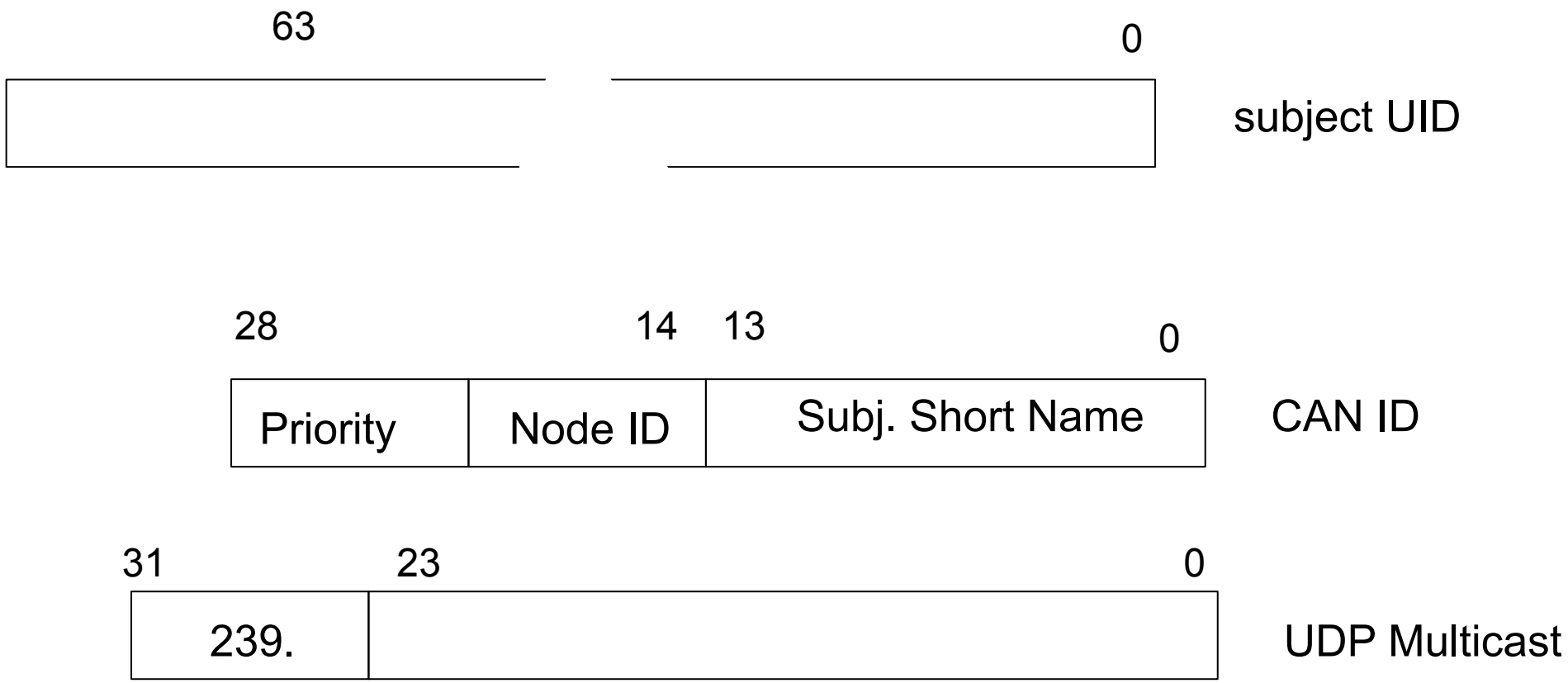


aus: M. Schulze: Adaptierbare ereignisbasierte Middleware für ressourcenbeschränkte Systeme, Disseration OVGU, 2011

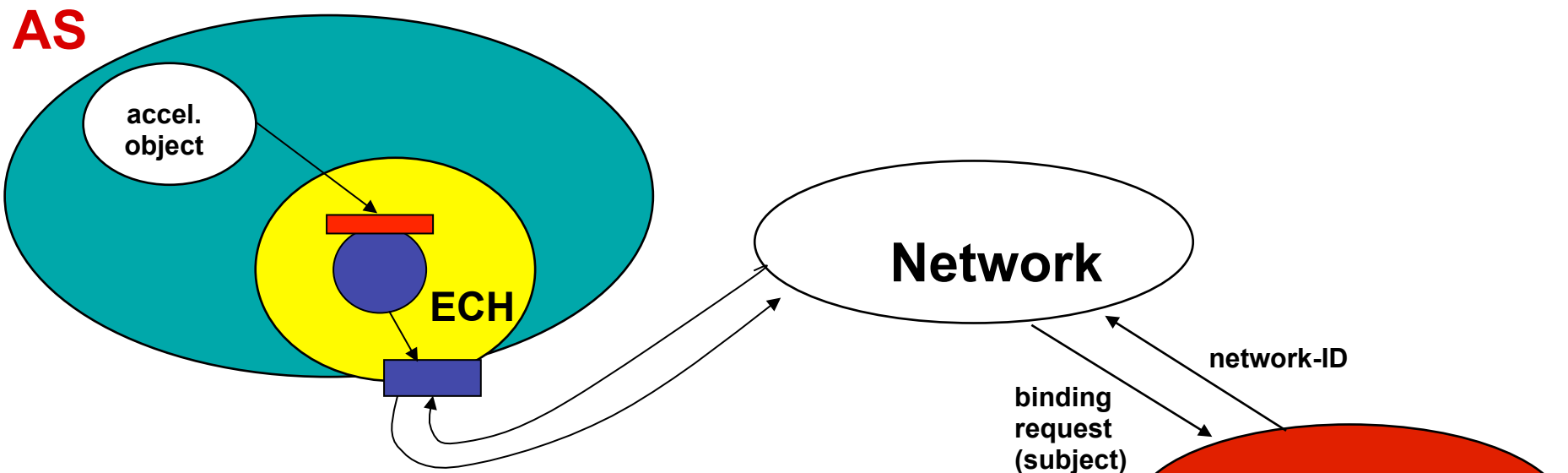
Local architecture

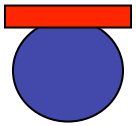


Mapping a subject to a network name



Communication architecture



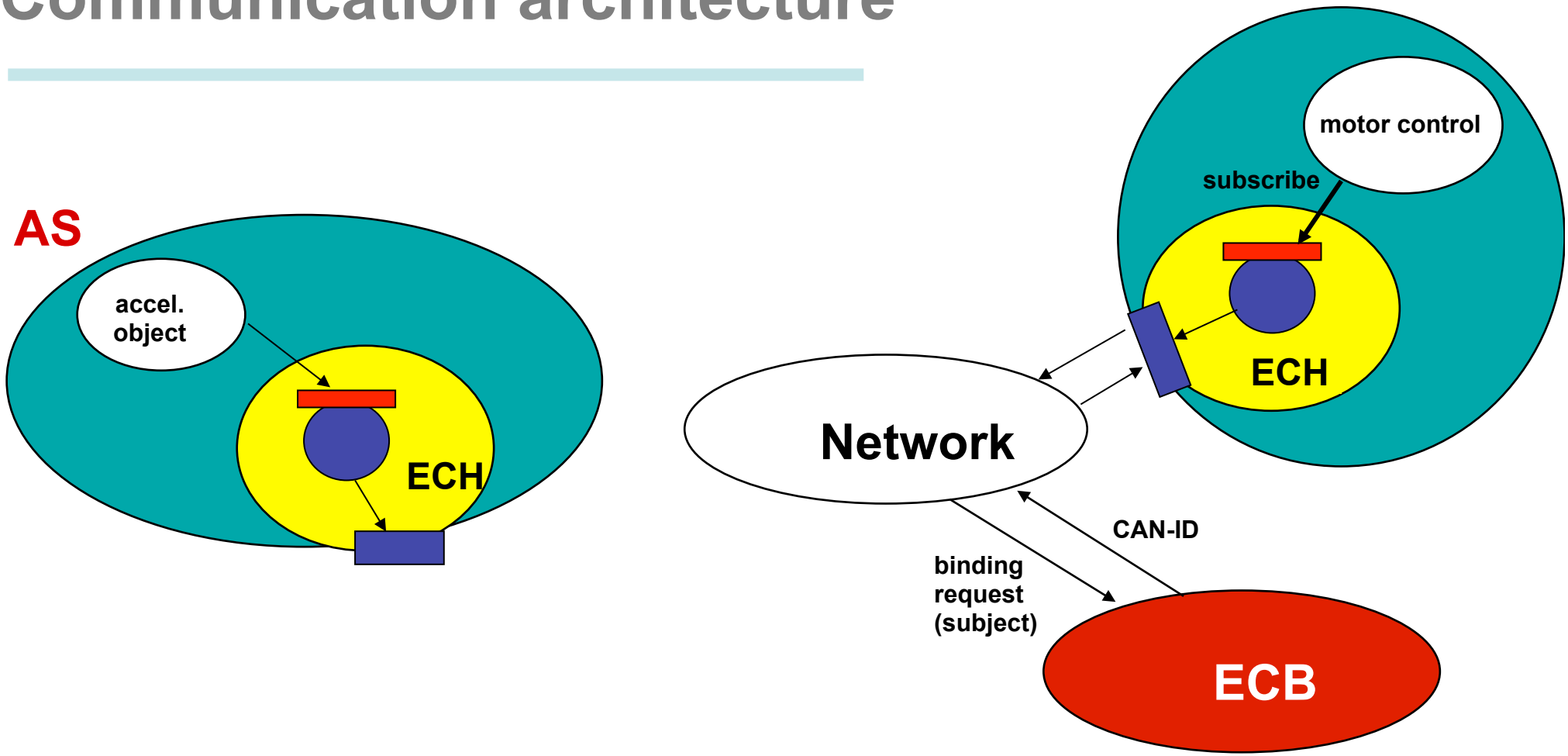


: Event Channel

ECH: Event Channel Handler

ECB: Event Channel Broker

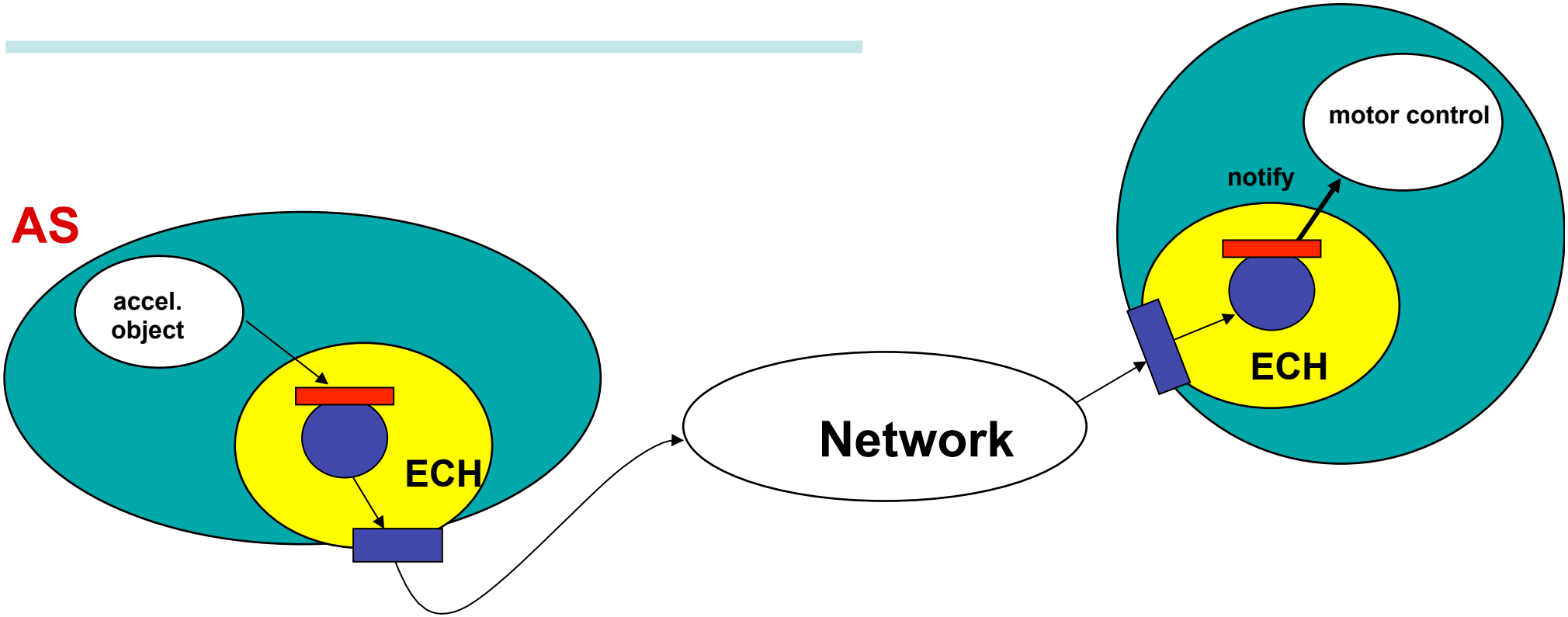
Communication architecture



ECH: Event Channel Handler

ECB: Event Channel Broker

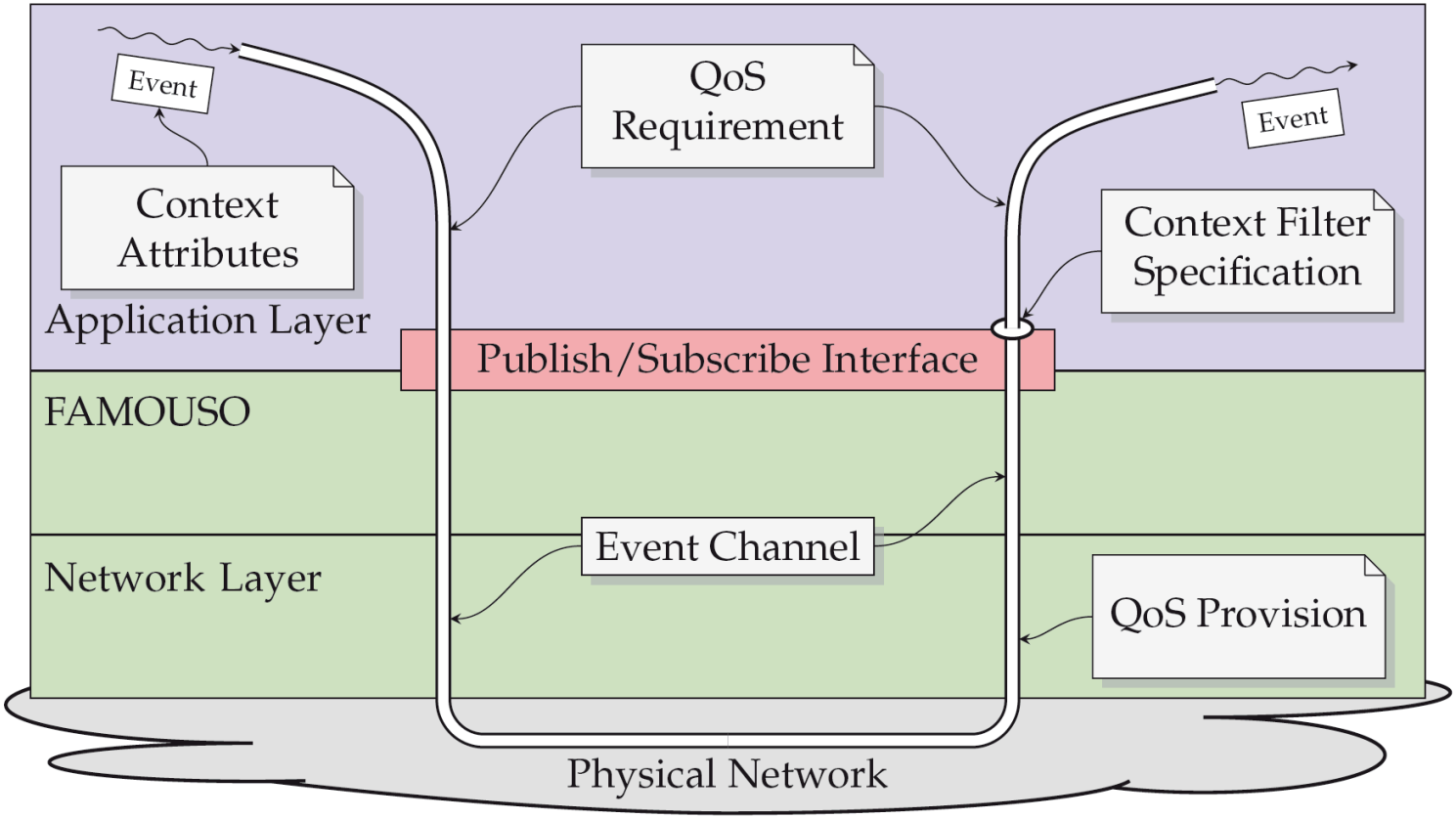
Communication architecture



ECH: Event Channel Handler

ECB: Event Channel Broker

Attribute, Filter und QoS Anforderungen in FAMOUSO



Eigenschaften von FAMOUSO

- Konfigurierbare Ereigniskanäle mit definierten QoS Eigenschaften
- Überprüfung der Anforderungen aus der Anwendung gegen die Eigenschaften der bereitgestellten (Netzwerk-) Hardware
- Spezifikations- und Filtersprachen, die effizient umgesetzt werden
- Themen- und Attributbasierte Filter
- Geschickte Ausnutzung der Netzwerkhardware zur themenbasierten Filterung
- Verteilte Architektur, keine zentrale Komponente während des Betriebs
- Broker nur in der Bindungsphase notwendig