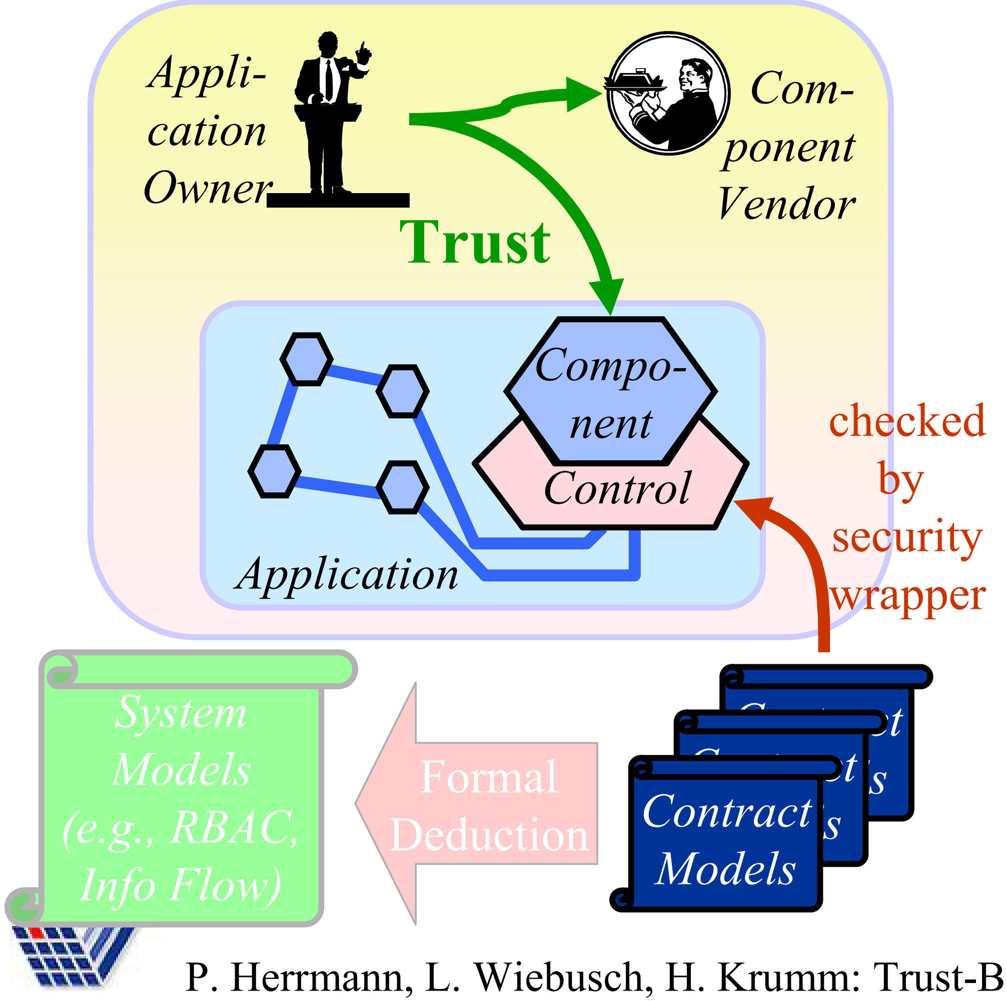
Trust-Based Runtime Monitoring of Distributed Component-Structured E-Commerce Software



Appli-cation Owner

Com-ponent Vendor

Trust

Compo-nent

Control

Application

checked by security wrapper

System Models (e.g., RBAC, Info Flow)

Formal Deduction

Contract Models

Peter Herrmann, Heiko Krumm
Computer Science Department
University of Dortmund

Lars Wiebusch
E-Plus Mobilfunk GmbH, Düsseldorf

Contents:

- Component-Structured Software
- Runtime Auditing
- Trust Management Support
- E-Procurement Application Example
- Concluding Remarks

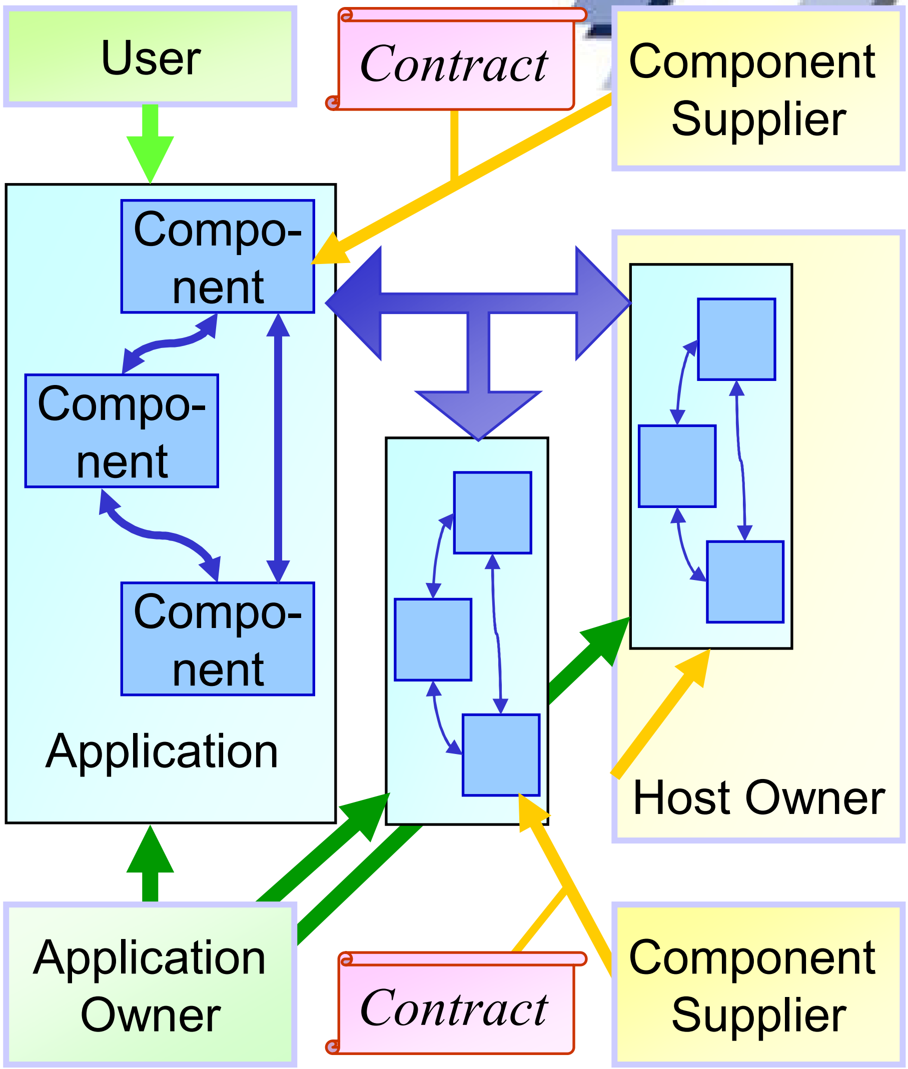
Component-Structured Software

Properties:

- Components:
  - Units of composition
  - Independent deployment
    - ➡ Support reuse
  - Independent development
  - Contractually specified interfaces
    - ➡ Only explicit context dependencies
    - ➡ Support configuration

Platforms:

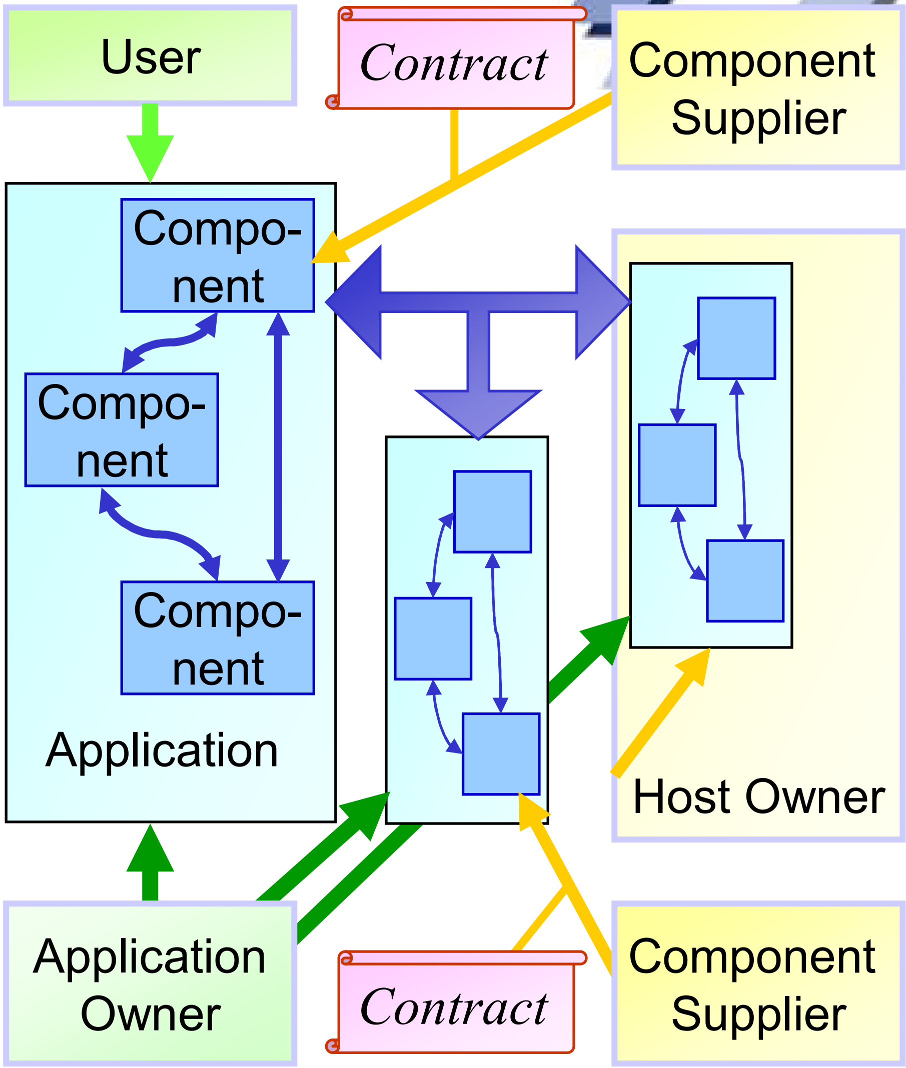
- Java Beans/EJB
- COM/DCOM/COM+
- CORBA component model



Component-Structured Software

Component security:

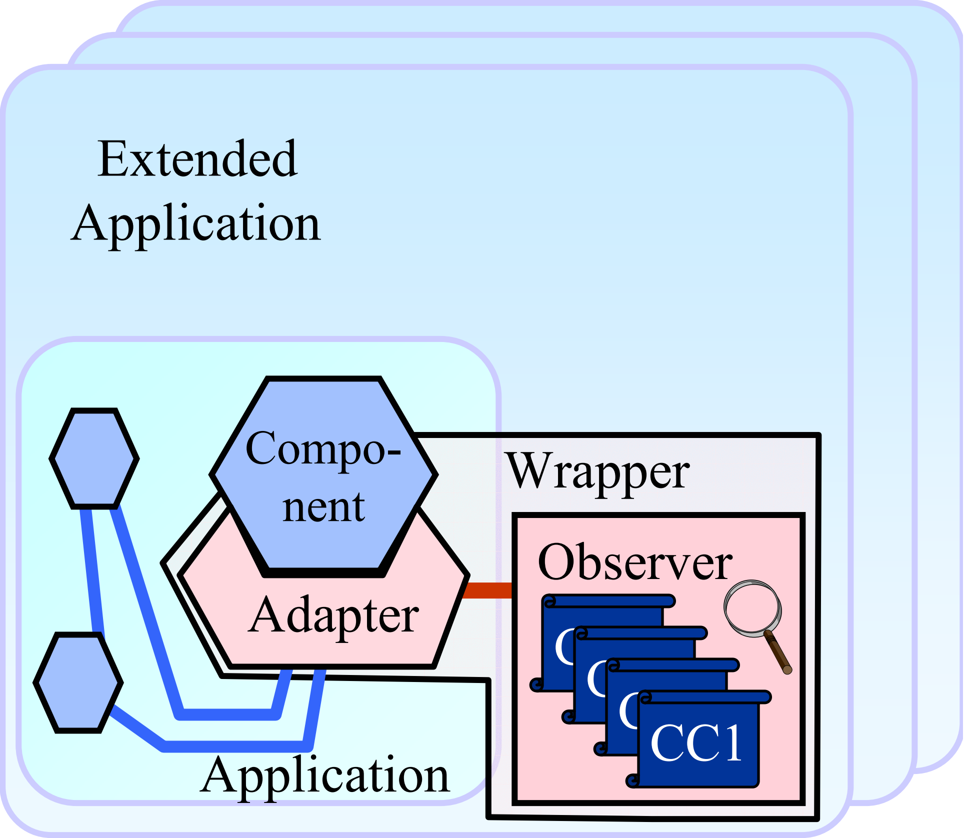
- Security objectives of distributed and mobile code systems
- New security objectives due to large number of principals:
  - **Protection of an application with respect to component attacks (confidentiality, integrity, availability,...)**
  - Protection of an application against a wrong coupling of components
  - Protection of component vendors against wrong incriminations
  - **Trust management**



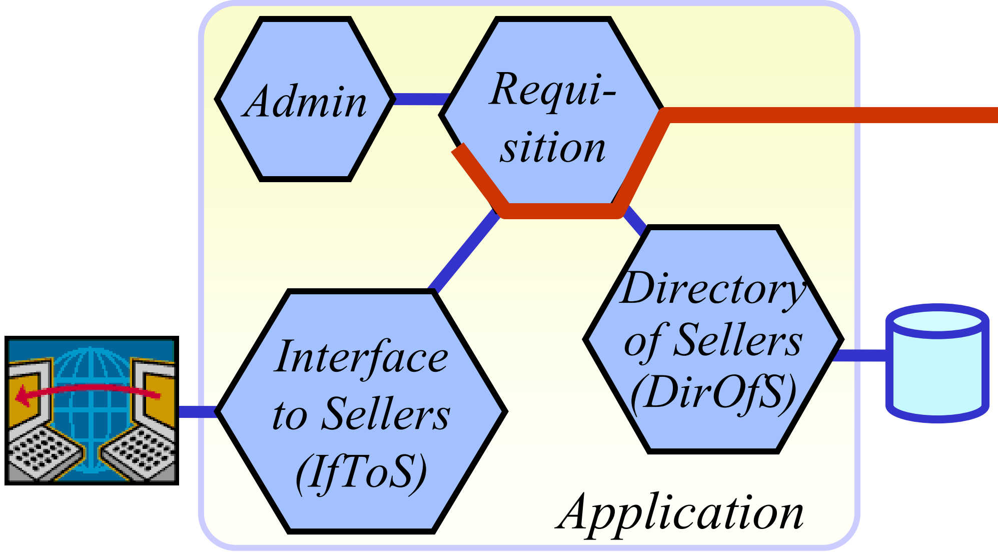
Runtime Auditing

Security Wrapper:

- Component contracts contain descriptions of security aspects
  - ➡ Model of legal interface actions
- Component in question is wrapped by an adapter
  - ➡ Interface traffic via adapter only
- Observer checks actual behavior against contract models
  - Adapter reports interface traffic
  - Observer checks interface event for compliance with the model
  - If an event is wrong,
    - » the component is blocked
    - » the application administrator is notified



Runtime Auditing



Contract:

*IfToS.ReqTender(**Saddr**)*
always only if after reading
S=DirOfS.read()
Saddr is member of the set ***S***

cTLA:

- Temporal Logic
- Based on TLA
- State-Transition-Systems
- Coupling by synchronously executed actions

```
process dfh2(SellerAdr : Any)
    var S : set of SellerAdr ;
    init ≡ S = { } ;
    actions
      DirOfS_read(sell : set of SellerAdr) ≡ S‘ = sell ;
      IfToS_ReqTender(Saddr : SellerAdr) ≡
        Saddr ∈ S ∧ S‘ = S ;
    end ;
```

Runtime Auditing

Component Contract Policy Patterns:

- ♦ Confidentiality:
  - ● Restriction of data flow
    - » **Data flow access**
    - » **Data flow history**
  - ● Deterministic behavior to prevent hidden channels
    - » **Hidden channel functional dependency**
    - » **Hidden channel enabling history**
    - » **Hidden channel exec. time**
- ♦ Integrity:
  - ● Constraining of interface events and their arguments
    - » **Integrity enabling condition**
    - » **Integrity enabling history**
- ♦ Availability:
  - ● Minimum waiting times to prevent denial-of-service attacks
    - » **Denial-of-service minimum waiting time**
    - » **Denial-of-service enabling history**
  - ● Maximum waiting times to prevent blocking of components
    - » **Blocking maximum waiting time**
    - » **Blocking enabling history**
- ♦ Non-repudiation:
  - ● Logging of events at a trusted third party service
    - » **Event logging**

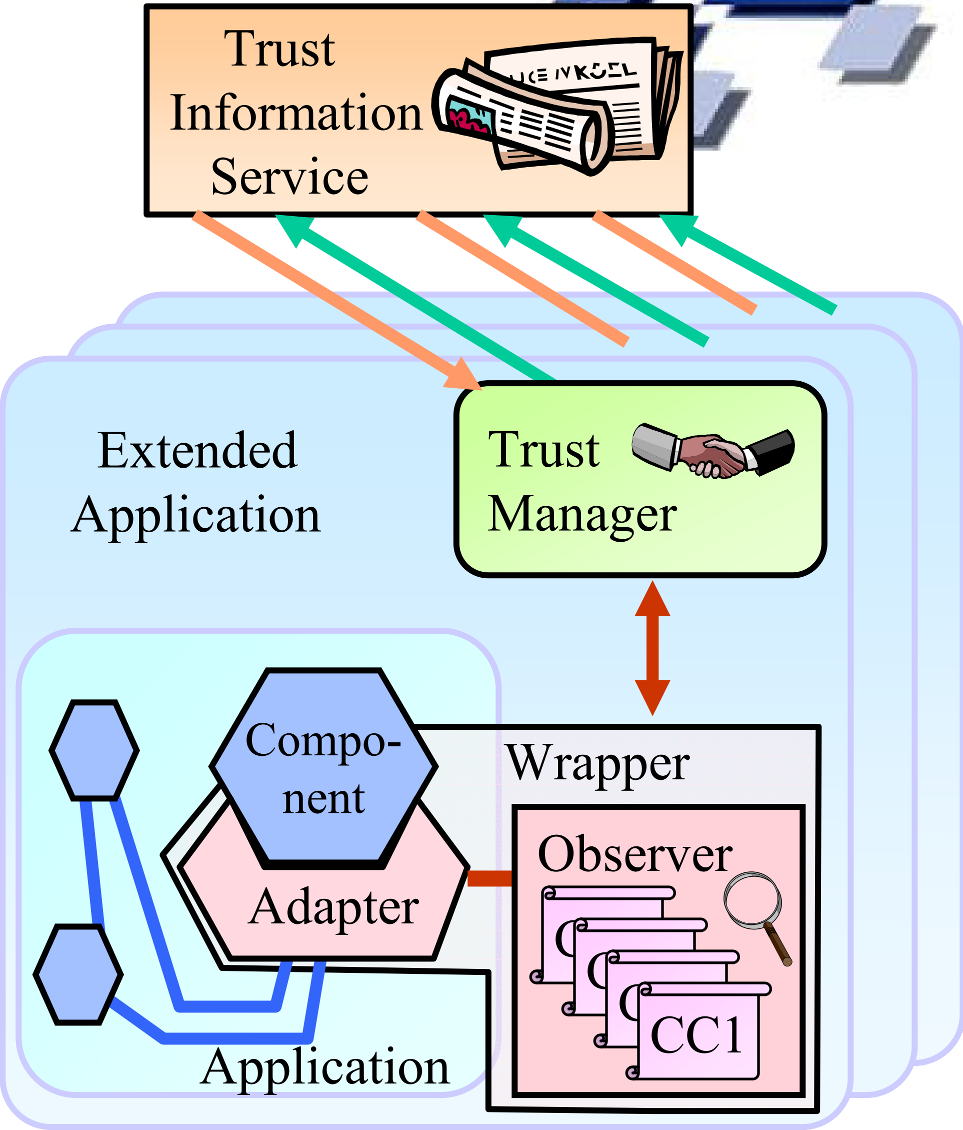
Trust Management Support

Trust Information Service:

- Collects good and bad evaluations on a component
- Calculates and offers trust values

Trust Manager:

- Varies enforcement depending on the current trust value:
  - Full observation
  - Spot checks
  - Remove wrapper
- Causes sealing of a component after an alarm message
- Replies inquiries from the Trust Information Service
- Notifies the Trust Information Service about severe violations

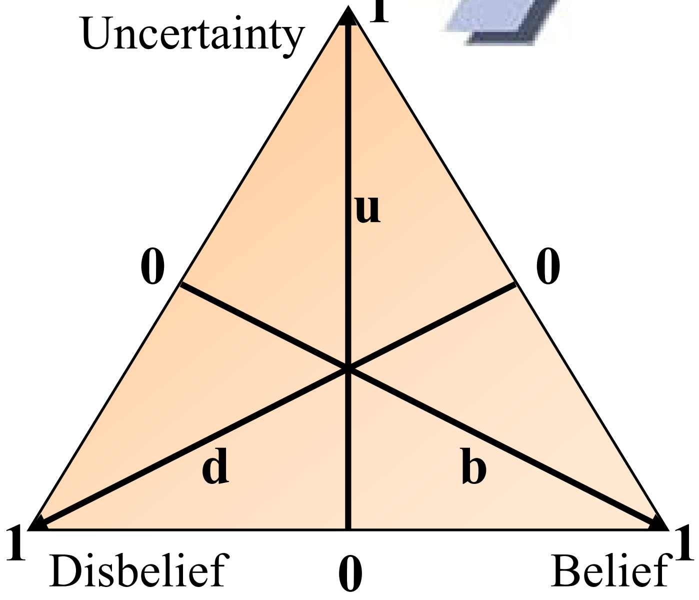


Trust Management Support

Trust modeling:

- Trust values:
  - Interval [0,1]
  - Triple <b, d, u>
    - b: belief
    - d: disbelief
    - u: uncertainty
    - ➡ b + d + u = 1
- Trust value determination
  - Calculation from the number of
    - positive experiences p
    - negative experiences n
  - Metrics:
    - Jøsang, Knapskog: liberal philosophy
    - Beth, Borcherding, Klein: unforgiving philosophy

- Opinion triangle (Jøsang)



- Metric of Jøsang, Knapskog:

$$b = \frac{p}{p+n+1} \quad d = \frac{n}{p+n+1} \quad u = \frac{1}{p+n+1}$$

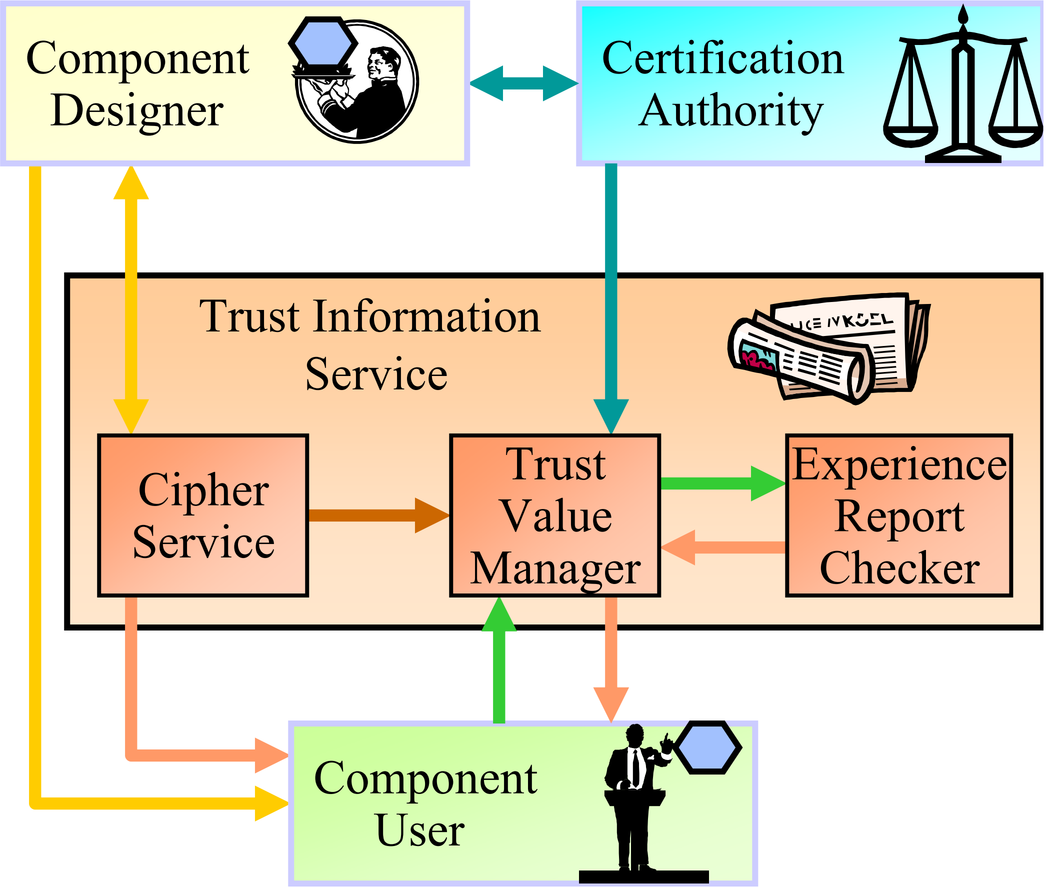
- Metric of Beth, Borcherding, Klein:

$$b = \begin{cases} 1-\alpha^{p}; & n = 0 \\ 0; & n > 0 \end{cases}$$

Trust Management Support

Trust Information Service:

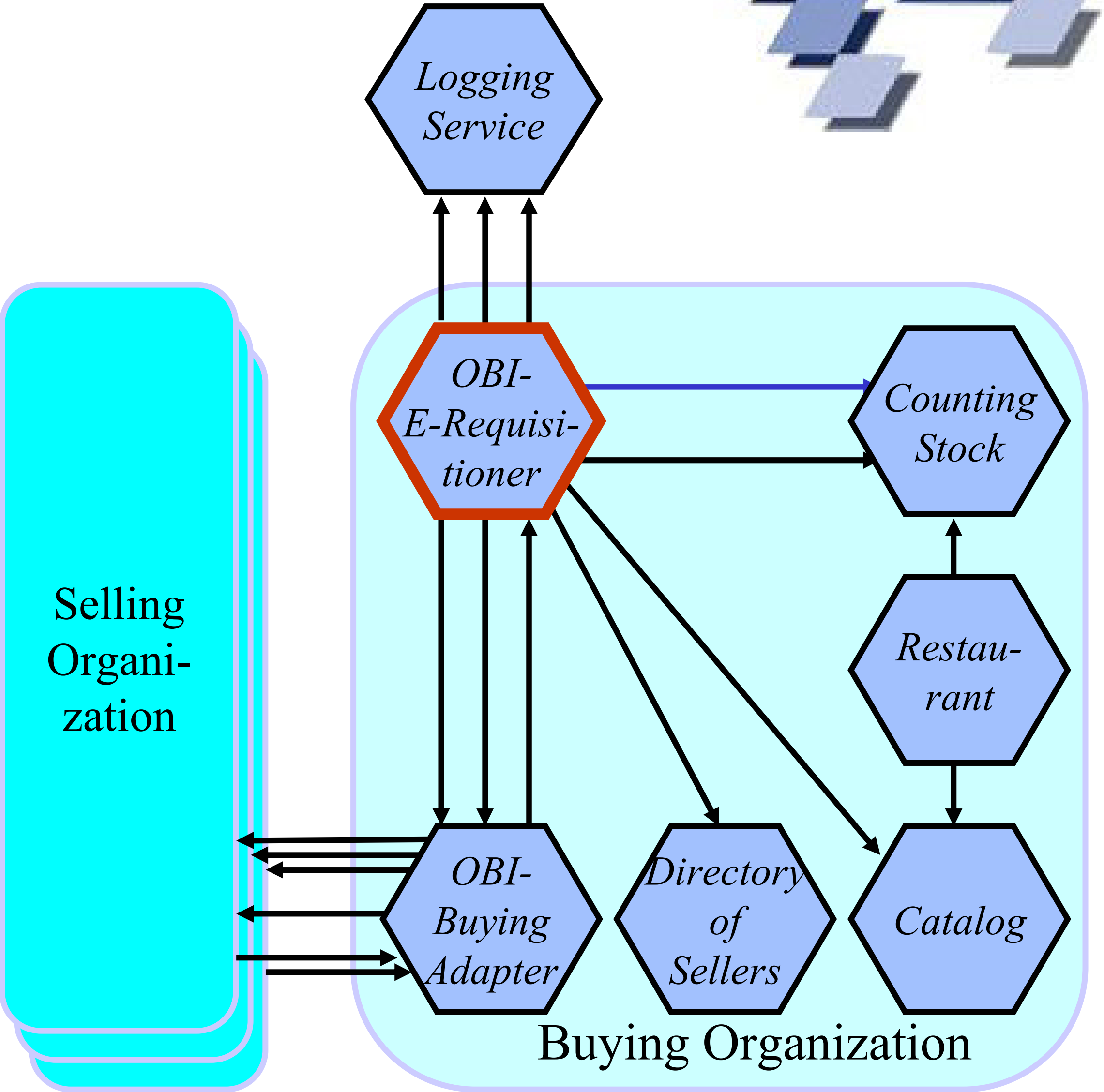
- Storage of
  - Component trust values
  - Recommendation trust values of component users
  - Trust values are stored based on ciphers
  - ➡ Privacy improvement
- Computation:
  - Experience reports are checked for validity
  - ➡ User's trust value
  - Component trust values are computed by means of the subjective logic
- Application:
  - Security wrapper control
  - Procurement decisions



E-Procurement Application Example

Application:

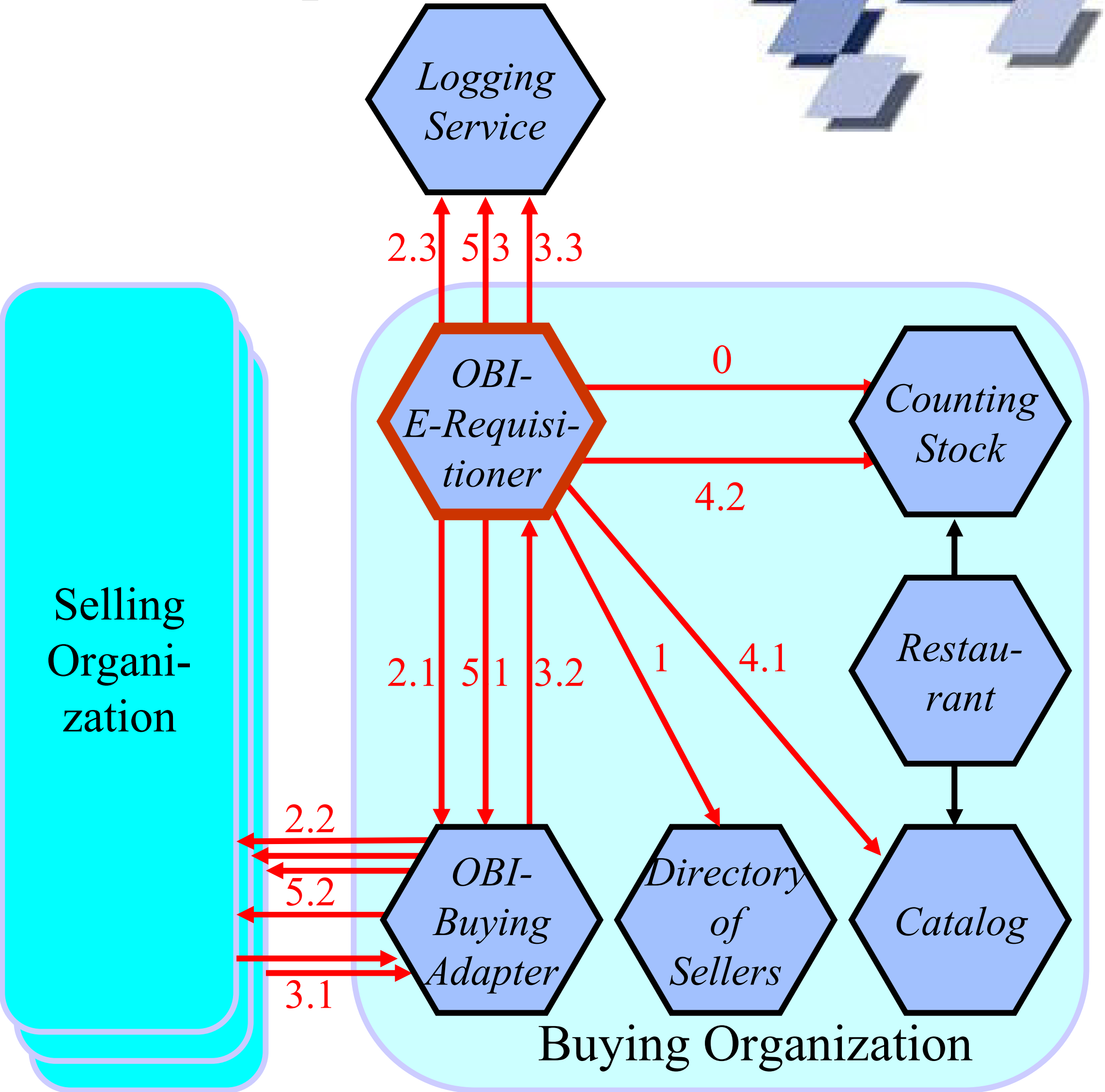
- Commodity management of fast-food restaurants
- OBI (Open Buying on the Internet)
  - Tender and Order formats
  - B2B model
- Component system:
  - *Restaurant, Counting Stock, Catalog,* Sellers adapted from the SalesPoint-Framework
  - *OBI-E-Requisitioner, OBI-Buying Adapter, Logging Service*



E-Procurement Application Example

OBI-B2B model:

0. Check counting stock
1. Ask buying organization for sellers
2. Request sellers for tenders
3. Receive tenders from sellers
4. Select winning seller and generate an order
5. Send order to the winning seller
6. Fulfill order
7. Pay order



E-Procurement Application Example

Critical Component:

- *OBI-E-Requisitioner*

  ➡ Security Policy Enforcement

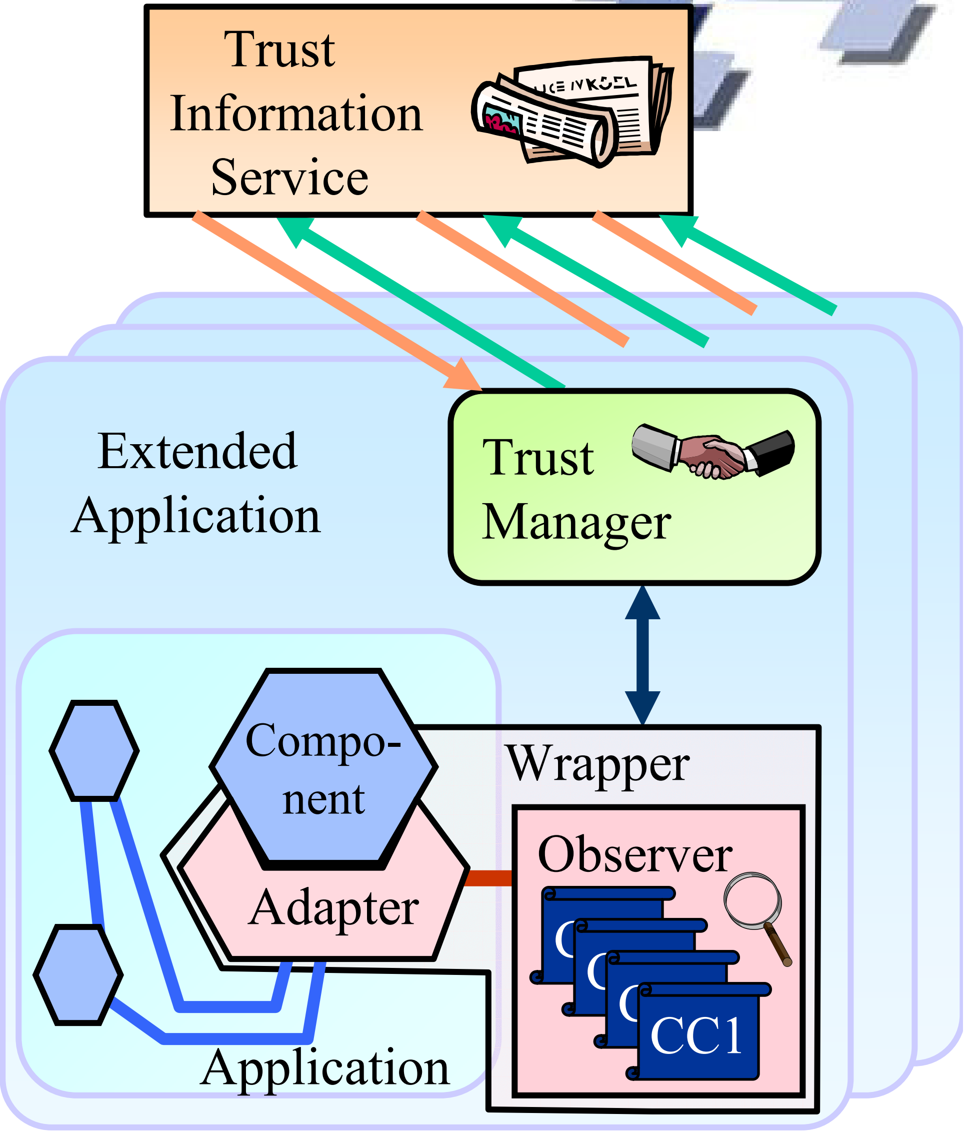
13 policies based on the patterns:

- Confidentiality (4 policies):
  - Relevant information is only forwarded to appropriate sellers
  - Hidden channels are not used to send competitor’s tenders
- Integrity (4 policies):
  - Relevant variables of the environment components are not altered
  - All selling organizations have a fair chance to win the order
  - The ordered amount is sensible
- Availability (4 policies):
  - Preventing denial-of-service attacks by demanding minimum waiting times between calls
  - Guaranteeing contemporary orders by demanding maximum waiting times for relevant steps
- Non-repudiation (1 policy):
  - Logging tender requests, tenders, and orders at the logging service

E-Procurement Application Example

Wrapper enforcement policies:

- Application security policy:
  - Highest security level: always full observation
  - Medium security level: Metric of Beth et al.;
    - spot checks: b > 0,9999 (7000 positive reports)
    - wrapper removed: b > 0,99999 (11600 positive reports)
  - Lowest security level: Metric of Jøsang, Knapskog;
    - spot checks: b > 0,99 (p ≥ 100 · n)
    - wrapper removed: b > 0,999 (p ≥ 1000 · n)



E-Procurement Application Example

Critical Component:

- *OBI-E-Requisitioner*

  ➡ Security Policy Enforcement

13 policies based on the patterns:

- Confidentiality (4 policies):
  - Relevant information is only forwarded to appropriate sellers
  - Hidden channels are not used to send competitor’s tenders
- Integrity (4 policies):
  - Relevant variables of the environment components are not altered
  - All selling organizations have a fair chance to win the order
  - The ordered amount is sensible
- Availability (4 policies):
  - Preventing denial-of-service attacks by demanding minimum waiting times between calls
  - Guaranteeing contemporary orders by demanding maximum waiting times for relevant steps
- Non-repudiation (1 policy):
  - Logging tender requests, tenders, and orders at the logging service

Runtime overhead:

- 5.4 % by run-time enforcement
- Reduction to 3.2 % by using trust management

Concluding Remarks

Introduced:

- ♦ Runtime auditing of components
- ♦ Trust management support

Other Application of Component Contract Models:

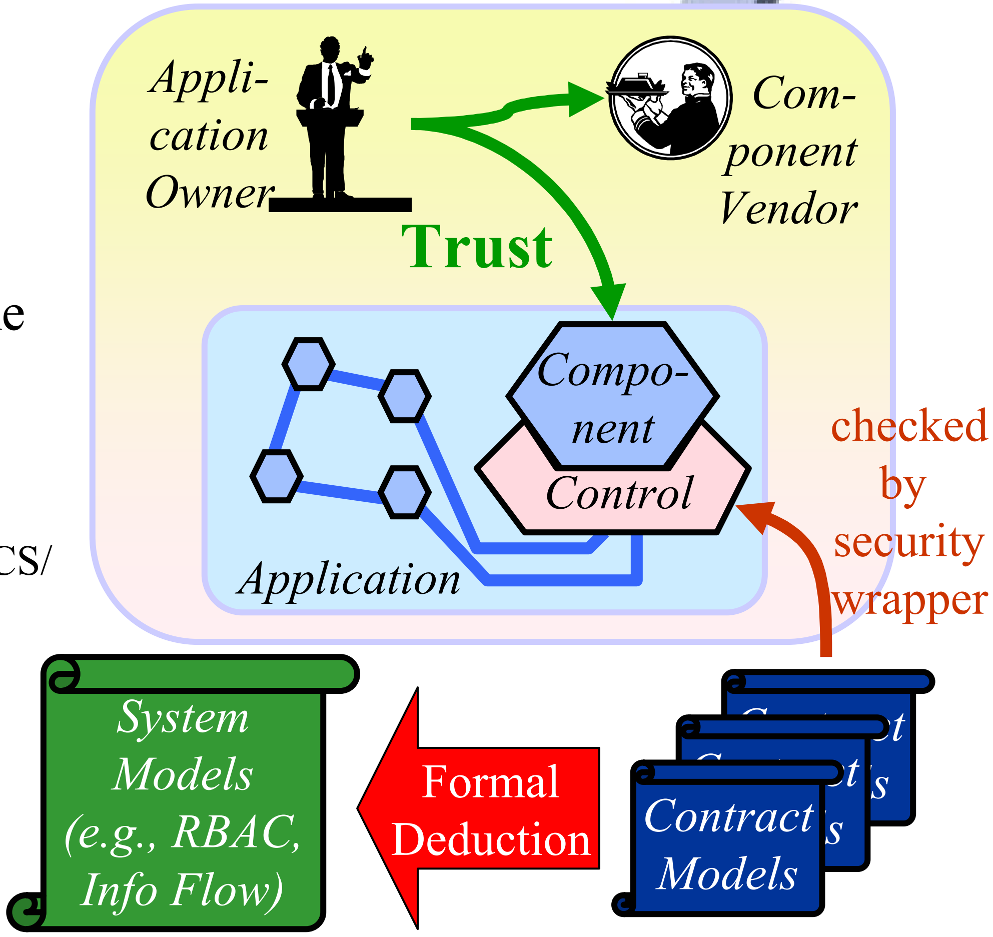
- ♦ Formal Verification at design time
  - ● Contract models fulfill global security models

Web-page:

ls4-www.cs.uni-dortmund.de/RVS/P-SACS/

To do:

- ♦ Runtime auditing:
  - ● UML models instead of cTLA
- ♦ Trust management support:
  - ● System risk analysis to define wrapper enforcement policies


Appli-
cation
Owner
Com-
ponent
Vendor
Trust
Compo-
nent
Control
Application
checked
by
security
wrapper
System
Models
(e.g., RBAC,
Info Flow)
Formal
Deduction
Contract's
Models