

# A Formal Model for Self-Adaptive and Self-Healing Organizations

Robrecht Haesevoets

Robrecht.Haesevoets@cs.kuleuven.be

*D. Weyns, T. Holvoet, W. Joosen  
KULeuven, Belgium*

**DistriNet**  
RESEARCH GROUP

KATHOLIEKE UNIVERSITEIT  
**LEUVEN**

# Overview

## **Dynamic Organizations and Roles**

- Agents: distributed setting - no central control
- Structure dynamic collaborations

## **Organization Middleware**

- Dynamic organizations as a reusable service

## **Formalization**

- Rigorous specification of
  - Organizational concepts
  - Middleware mechanisms

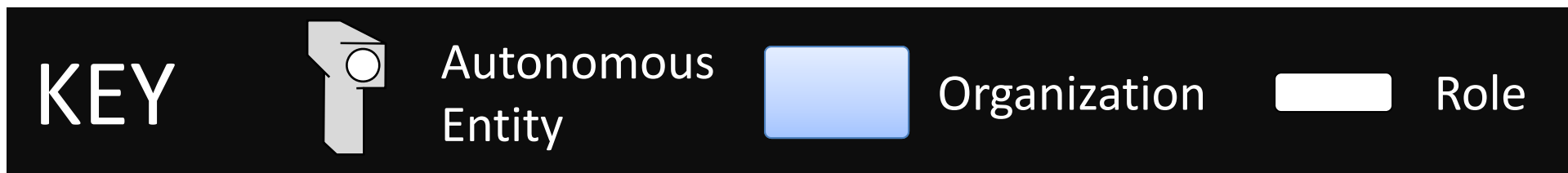
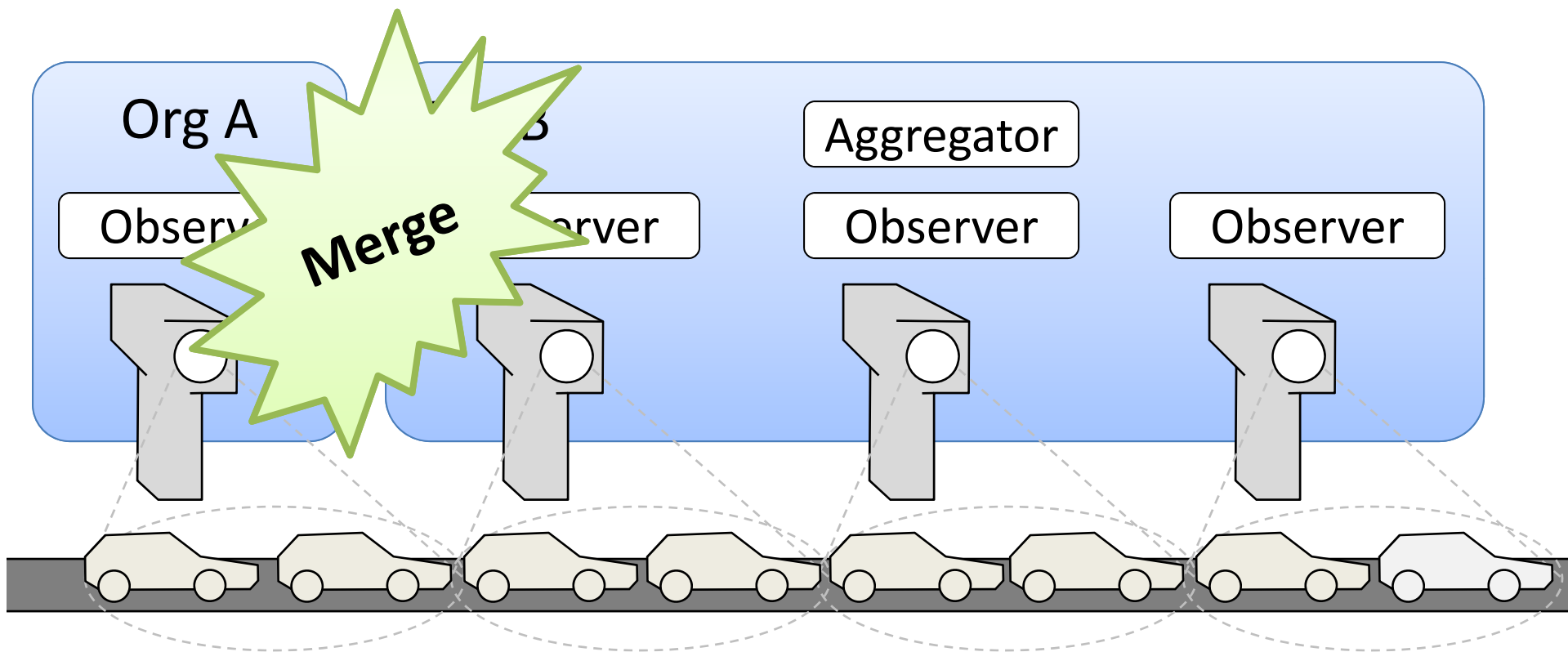
# Dynamic Organizations

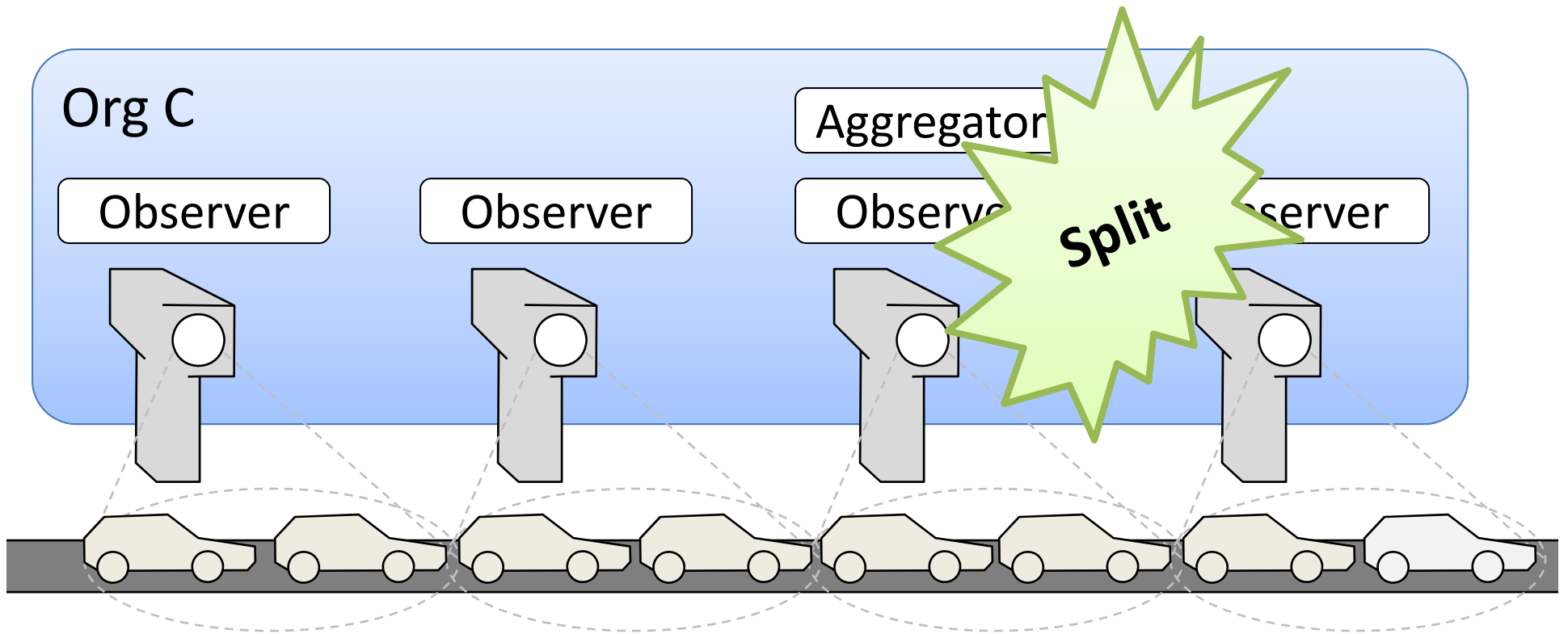


Self-Adaptive

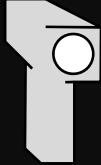
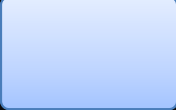



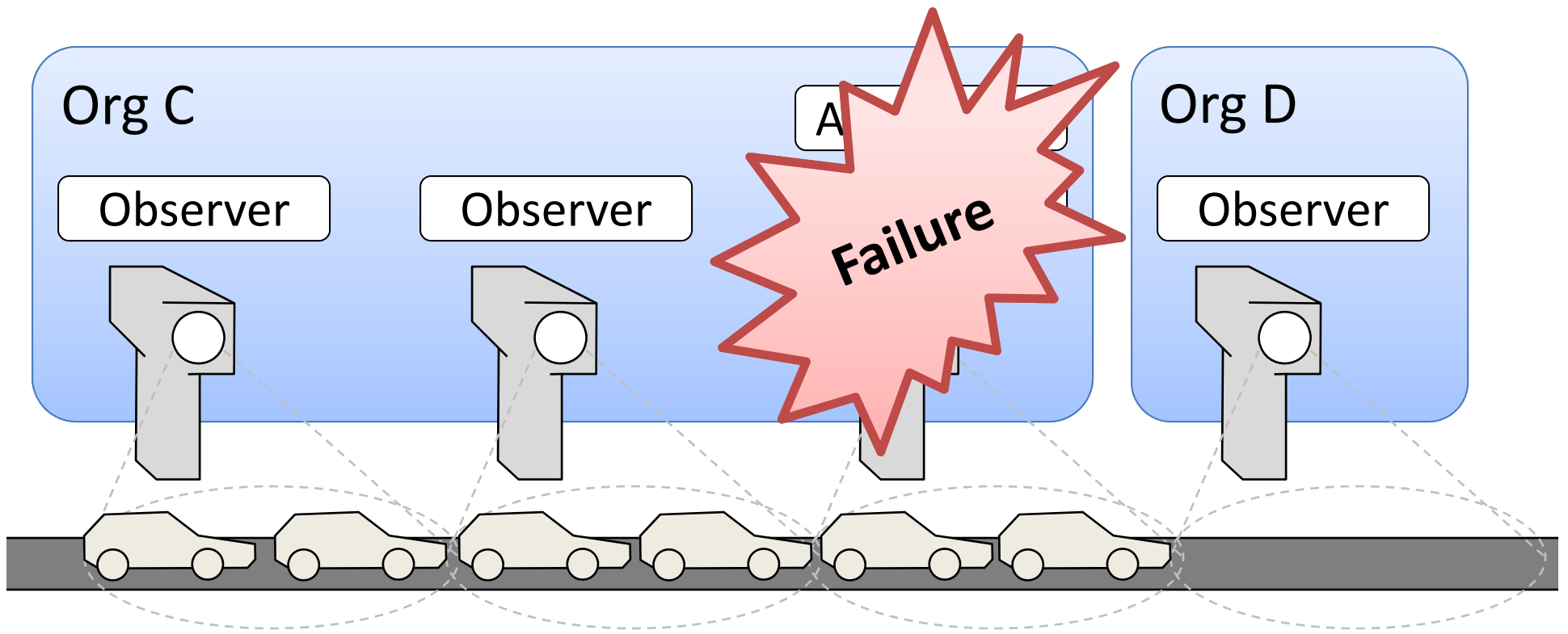
Self-Healing



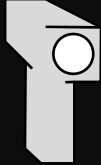




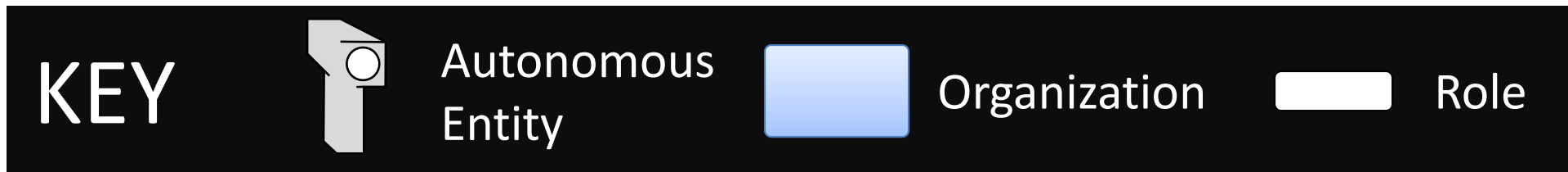
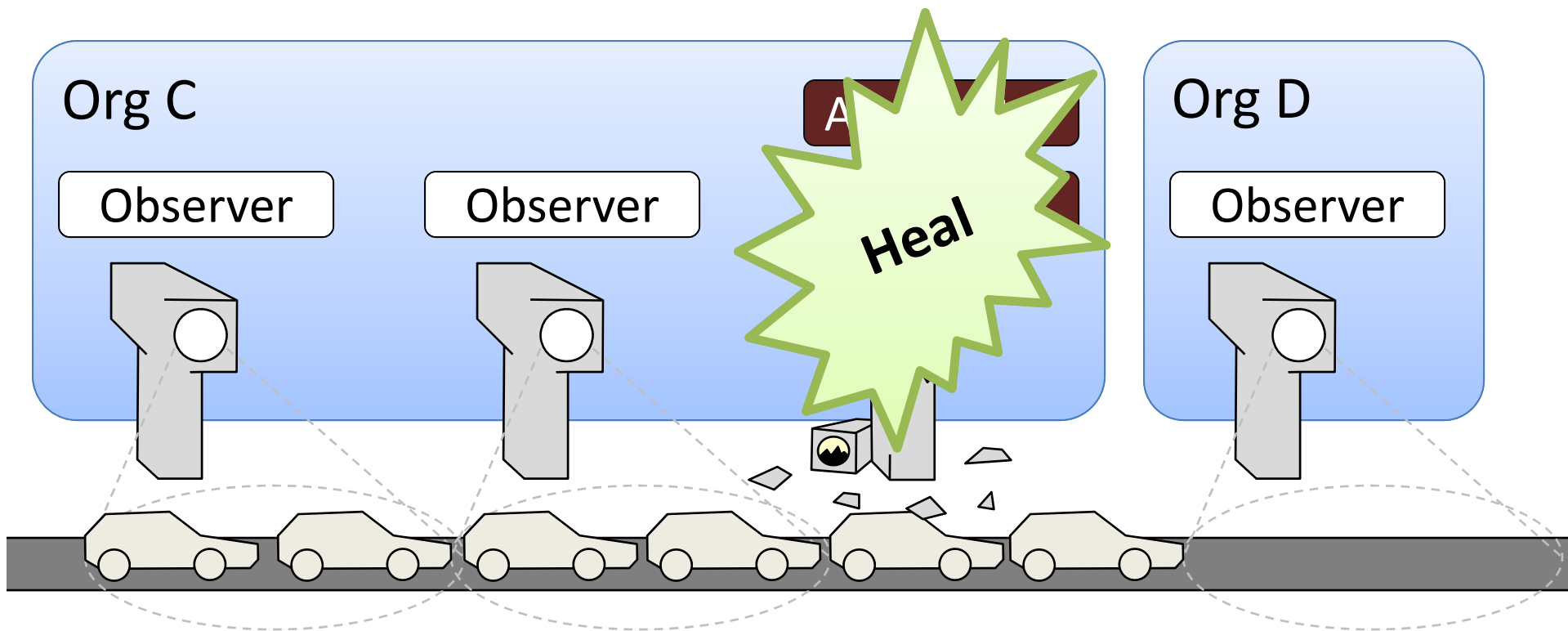
**KEY**

-  Autonomous Entity
-  Organization
-  Role

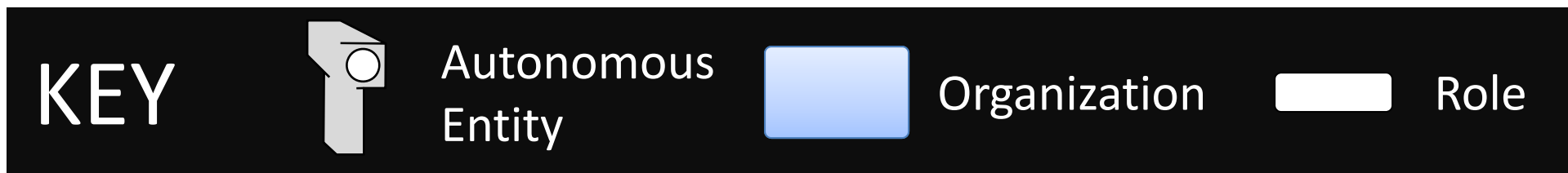
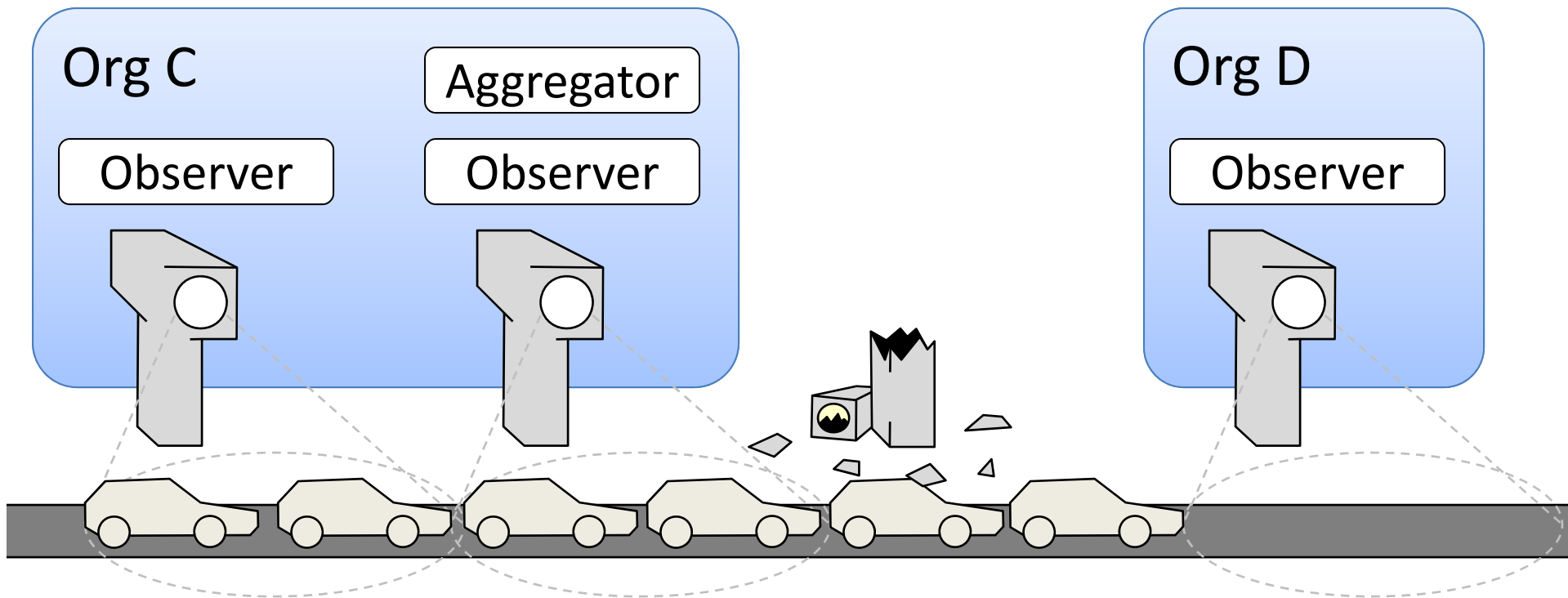


**KEY**

-  Autonomous Entity
-  Organization
-  Role

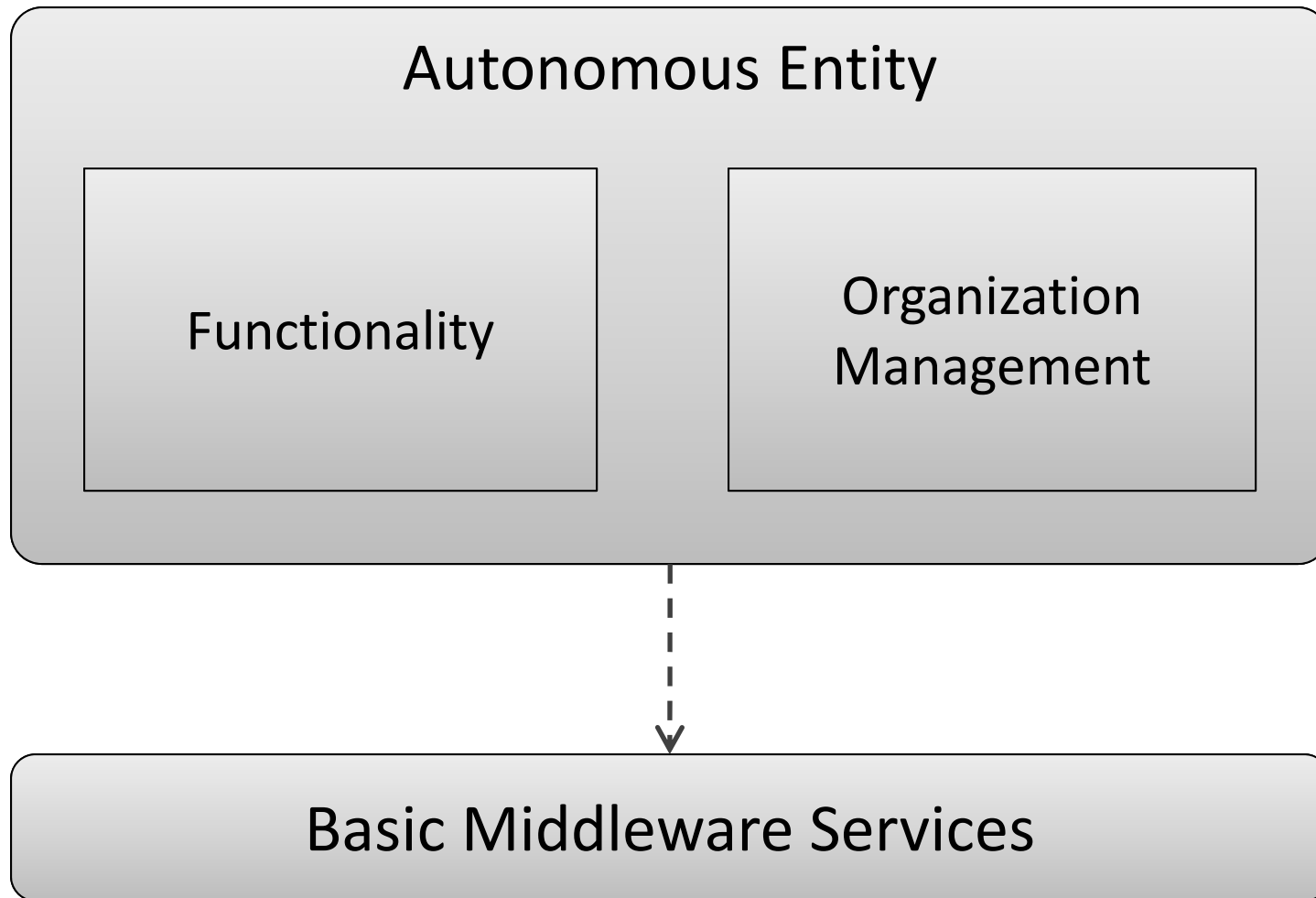






# Organization Middleware

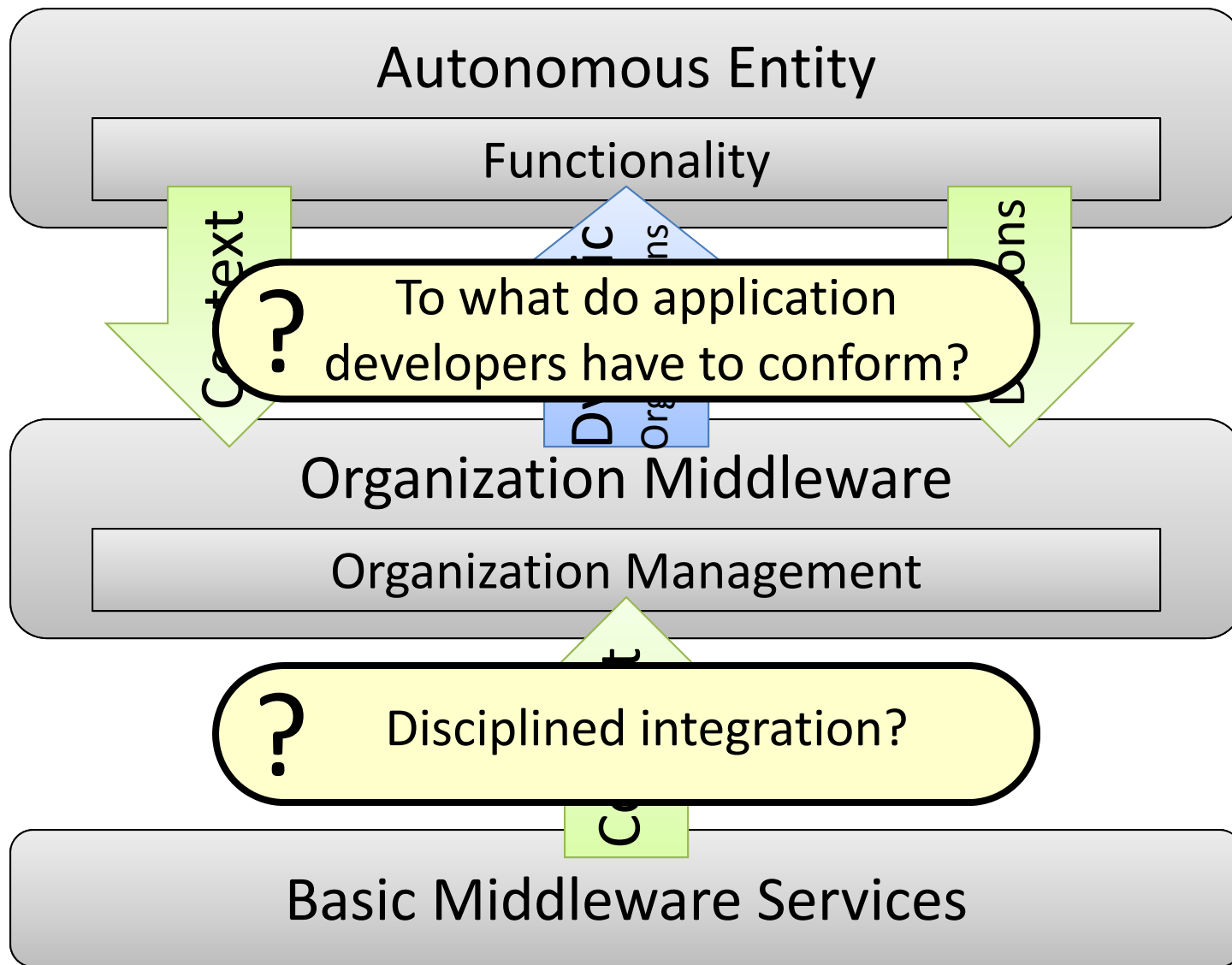
*Supporting  
Dynamic Collaborations*



\* V. Dignum et al. - *Formal Specification of Interaction in Agent Societies* – LNCS, 2003

\* J. Ferber et al. - *AGRE: Integrating environments with organizations* – LNCS, 2005

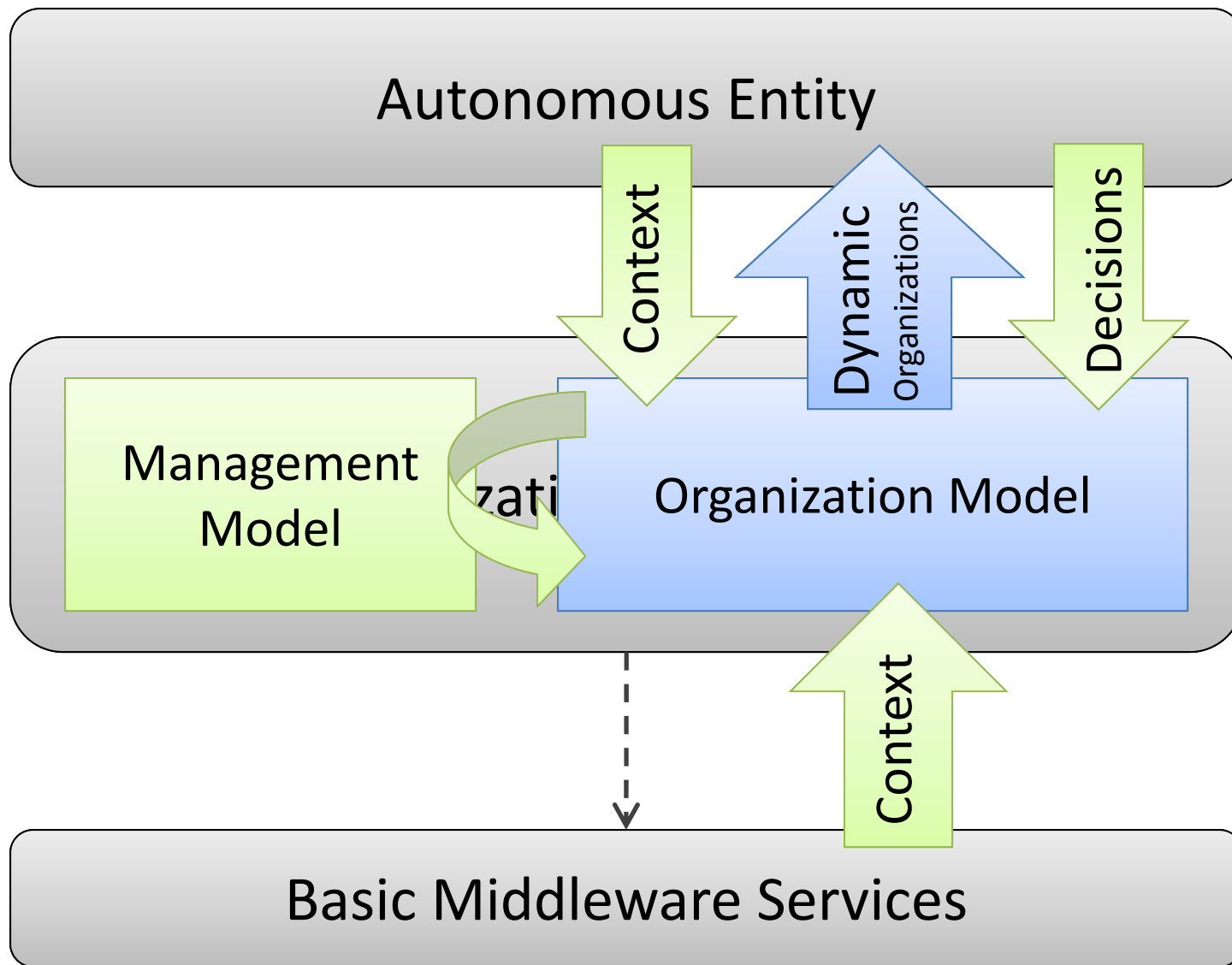




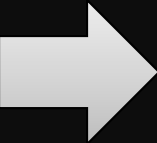


**KEY** Layer Module Allowed to use Information Flow

# Formalization

- \* *V. Dignum et al. - Formal Specification of Interaction in Agent Societies – LNCS, 2003*
- \* *M. d’Inverno and M. Luck - Understanding Agent Systems – Springer Verlag, 2004*
- \* *J. Jacky - The Way of Z - Cambridge University Press, 1997*
- \* *CZT, 2008 - Community Z Tools - <http://czt.sourceforge.net/>*



**KEY**  Layer  Allowed to use  Information Flow

# Organization Model

## AGENT

$name_a : AGENTNAME$

$context_a : AGENTCONTEXT$

$capabilities_a : \mathbb{P} CAPABILITY$

$roleContracts_a : \mathbb{P} ROLECONT$

$\forall rc : roleContracts_a \bullet rc.agentName = name_a$

## ORG

$name_o : ORGNAME$

$context_o : ORGCONTEXT$

$roleContracts_o : \mathbb{P} ROLECONT$

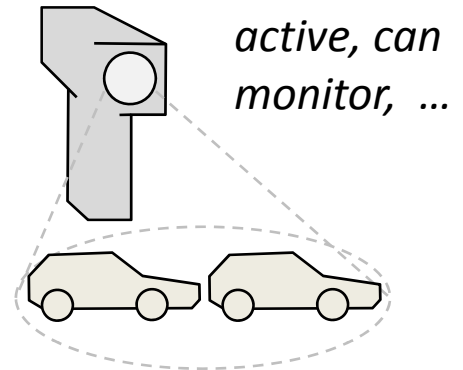
$rolePositions_o : \mathbb{P} ROLEPOS$

$\forall rp : rolePositions_o \bullet rp.orgName = name_o$

$\forall rc : roleContracts_o \bullet$

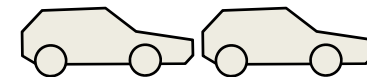
$rc.rolePosition.orgName = name_o$

Observer



Org A

Observer





*ORGANIZATIONMOD*

*agents* :  $\mathbb{P} \text{AGENT}$

*organizations* :  $\mathbb{P} \text{ORG}$

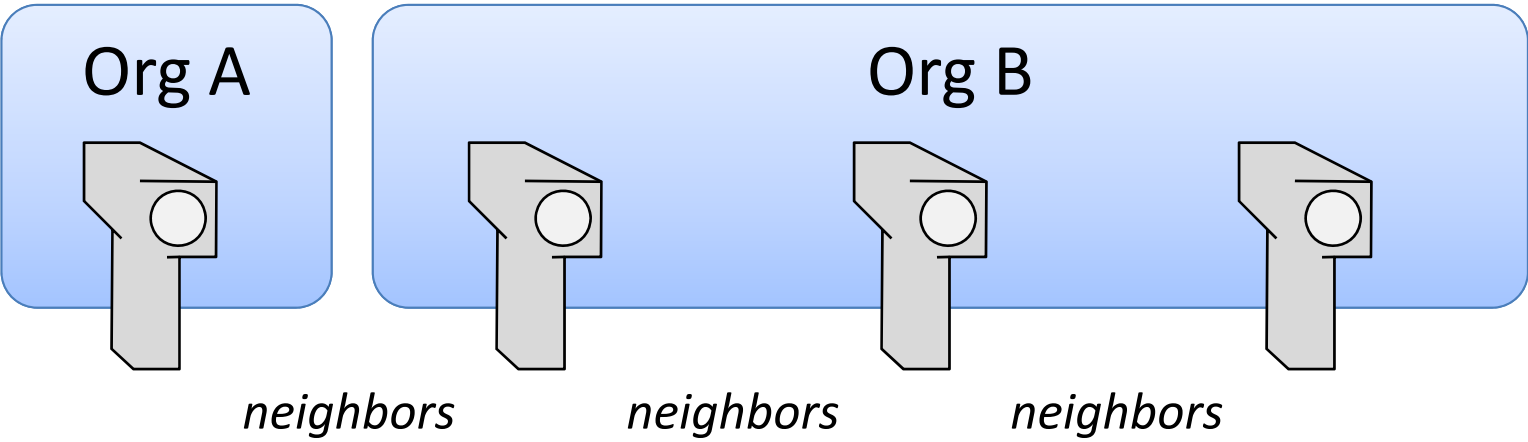
*contextRelations<sub>a</sub>* : *CONTEXTRELATIONS<sub>a</sub>*

*contextRelations<sub>o</sub>* : *CONTEXTRELATIONS<sub>o</sub>*

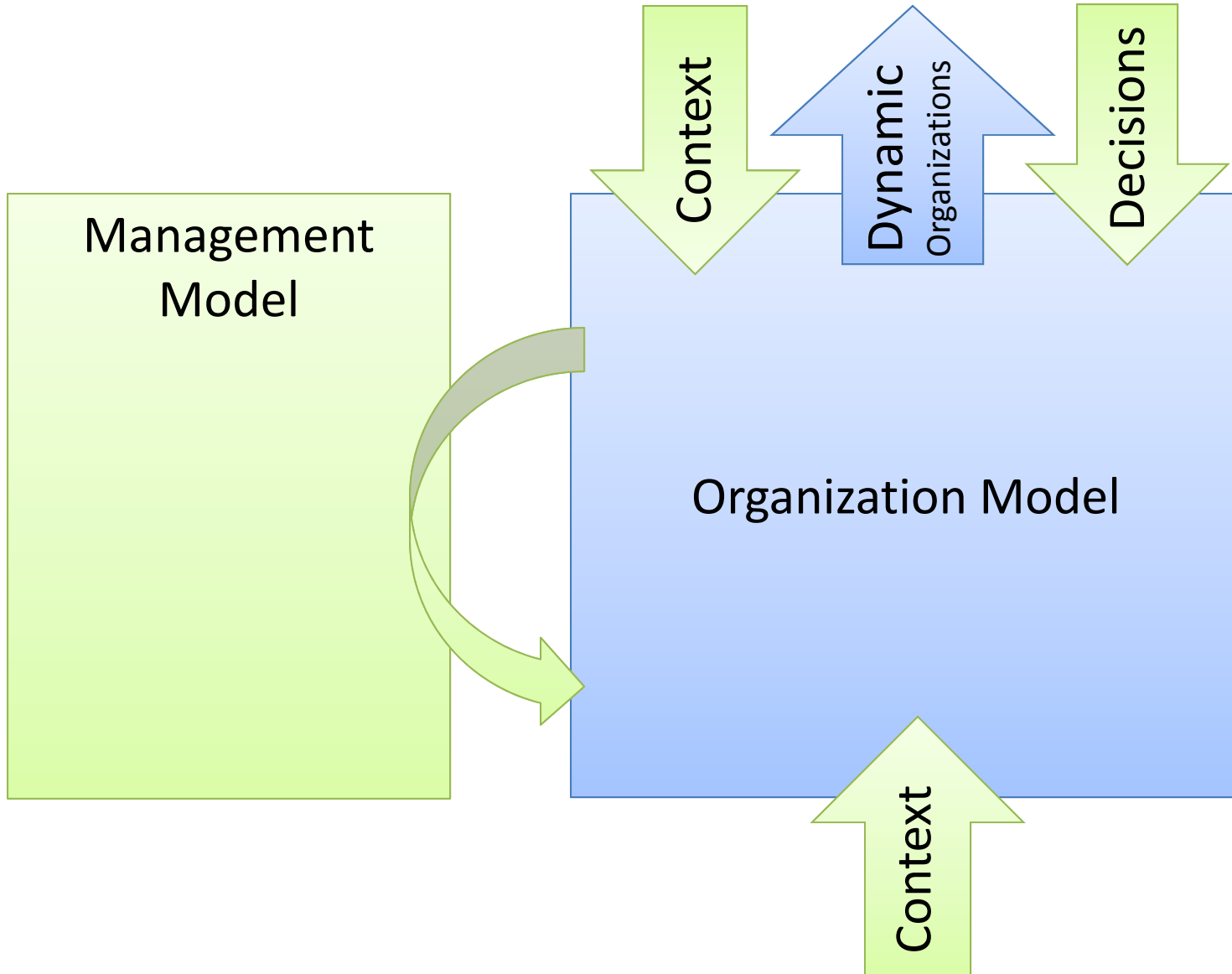
*uniquenames<sub>o</sub>* :  $\mathbb{P} \text{ORGNAME}$

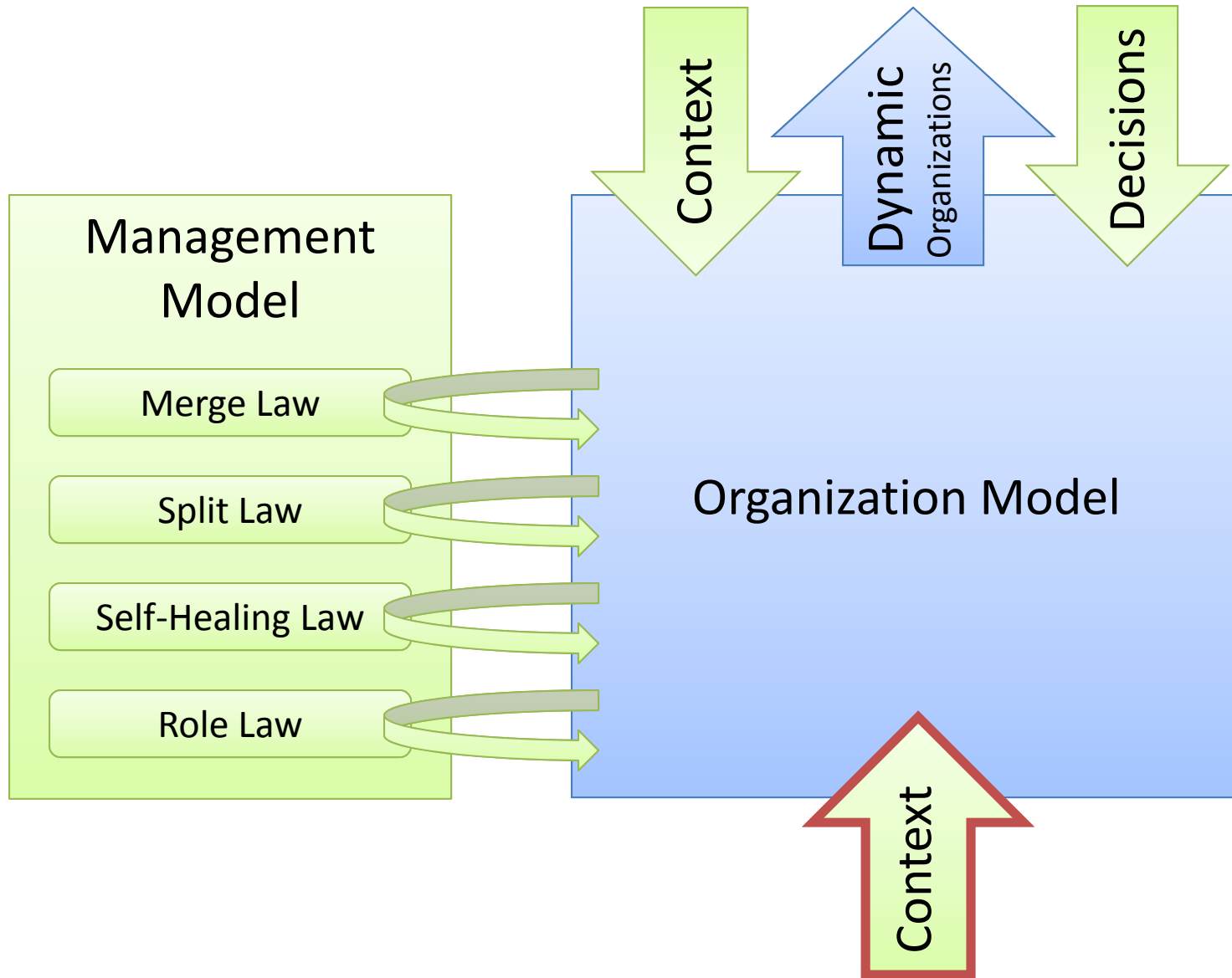
*uniquenames<sub>o</sub>* = {*n* : *ORGNAME* |  
     $\forall o : \text{organizations} \bullet n \neq o.\text{name}_o$ }  
 $\forall x, y : \text{agents} \bullet x.\text{name}_a = y.\text{name}_a \Leftrightarrow x = y$   
 $\forall x, y : \text{organizations} \bullet$   
     $x.\text{name}_o = y.\text{name}_o \Leftrightarrow x = y$

Organization Model



# Management Model





*CapabilityUpdate*

$\Delta$  *ORGANIZATIONMOD*

*agentname?* : <sup>*EL*</sup> *AGENTNAME*

*capabilities?* :  $\mathbb{P}$  *CAPABILITY*

$\exists oa : agents \bullet oa.name_a = agentname? \wedge$

$\exists ua : AGENT \bullet ua.name_a = oa.name_a \wedge$

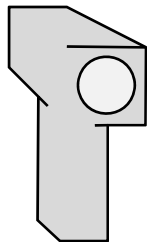
$ua.capabilities_a = capabilities? \wedge$

$ua.roleContracts_a = oa.roleContracts_a \wedge$

$ua.context_a = oa.context_a \wedge$

$organizations' = organizations \wedge$

$agents' = agents \setminus \{oa\} \cup \{ua\}$



*active, can  
monitor, can  
send, ...*

### CapabilityUpdate

$\Delta$  ORGANIZATIONMOD

FI

agentname? : AGENTNAME

capabilities? :  $\mathbb{P}$  CAPABILITY

$\exists oa : agents \bullet oa.name_a = agentname? \wedge$

$\exists ua : AGENT \bullet ua.name_a = oa.name_a \wedge$

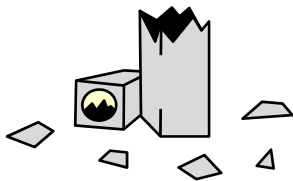
$ua.capabilities_a = capabilities? \wedge$

$ua.roleContracts_a = oa.roleContracts_a \wedge$

$ua.context_a = oa.context_a \wedge$

$organizations' = organizations \wedge$

$agents' = agents \setminus \{oa\} \cup \{ua\}$



broken, can't  
monitor, can't  
send, ...

### AGENT

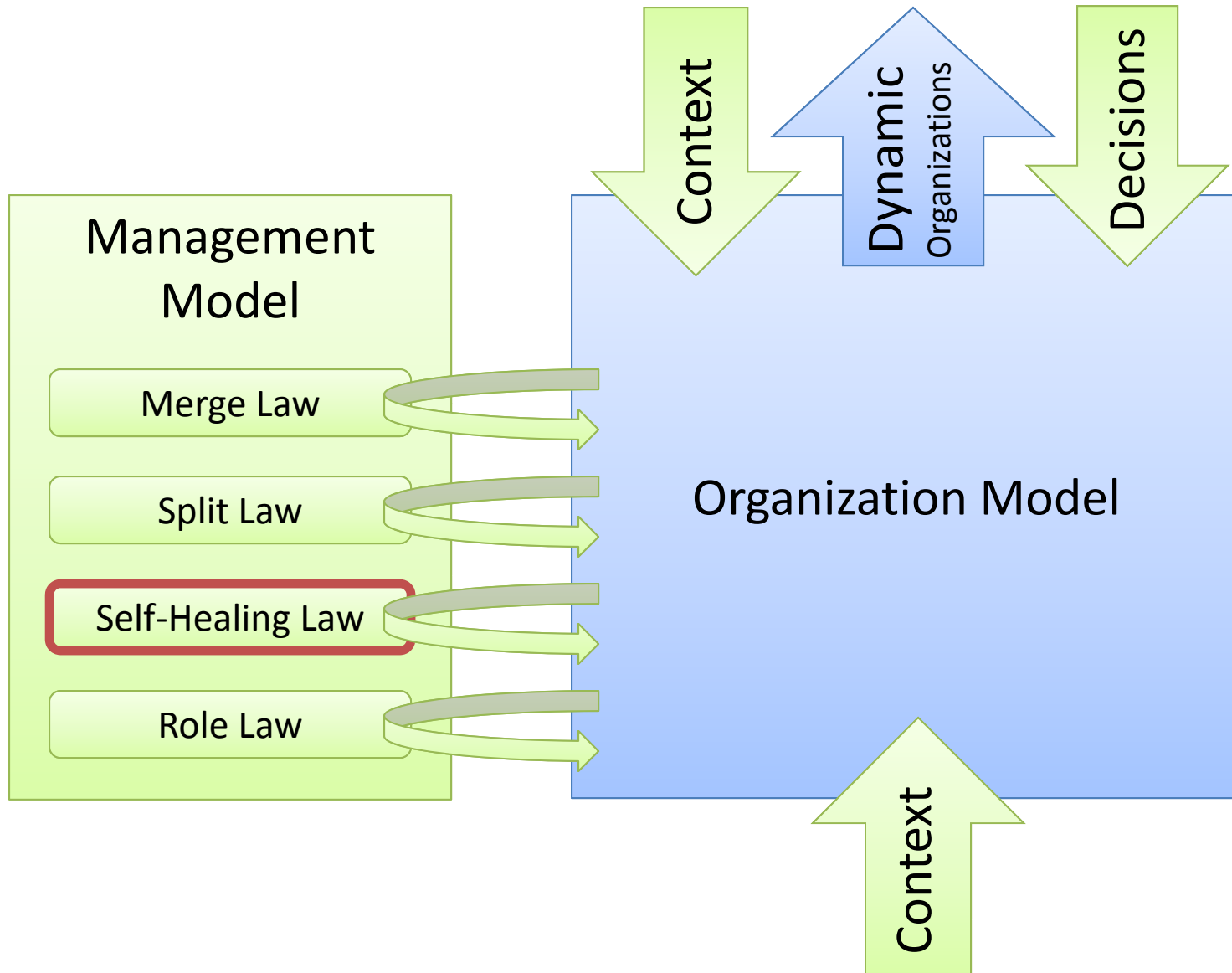
$name_a : AGENTNAME$

$context_a : AGENTCONTEXT$

$capabilities_a : \mathbb{P} CAPABILITY$

$roleContracts_a : \mathbb{P} ROLECONT$

$\forall rc : roleContracts_a \bullet rc.agentName = name_a$



SelfHealingLaw

$\Delta$  ORGANIZATIONMOD

$agent? : AGENT^{EL}$

$\exists invalidContracts : \mathbb{P} ROLECONT \bullet invalidContracts = invalidroleContracts(agent?) \wedge invalidContracts \neq \emptyset \wedge$

$\exists uagent : AGENT \bullet uagent = updateAgent_{contracts}(agent?, agent?.roleContracts_a \setminus invalidContracts) \wedge$

$\exists affecteddorgs : \mathbb{P} ORG \bullet affecteddorgs = \{o : organizations \mid \exists rc : invalidContracts \bullet rc \in o.roleContracts_o\} \wedge$

$\exists uorgs : \mathbb{P} ORG \bullet uorgs = \{uo : ORG \mid \exists ao : affecteddorgs \bullet$

$uo.name_o = ao.name_o \wedge uo.context_o = \{a : AGENT \mid a \text{ activeIn } uo \bullet a.context_a\} \wedge$

$uo.roleContracts_o = ao.roleContracts_o \setminus invalidContracts \wedge uo.rolePositions_o = ao.rolePositions_o\} \wedge$

$organizations' = organizations \setminus affecteddorgs \cup uorgs \wedge$

$agents' = agents \setminus \{agent?\} \cup \{uagent\}$

AGENT

$name_a : AGENTNAME$

$context_a : AGENTCONTEXT$

$capabilities_a : \mathbb{P} CAPABILITY$

$roleContracts_a : \mathbb{P} ROLECONT$

$\forall rc : roleContracts_a \bullet rc.agentName = name_a$

ORG

$name_o : ORGNAME$

$context_o : ORGCONTEXT$

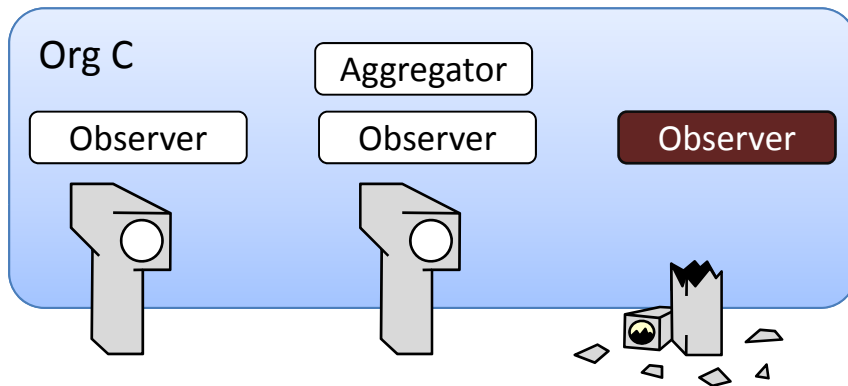
$roleContracts_o : \mathbb{P} ROLECONT$

$rolePositions_o : \mathbb{P} ROLEPOS$

$\forall rp : rolePositions_o \bullet rp.orgName = name_o$

$\forall rc : roleContracts_o \bullet$

$rc.rolePosition.orgName = name_o$





SelfHealingLaw

$\Delta$  ORGANIZATIONMOD

$agent? : AGENT^{EL}$

$\exists invalidContracts : \mathbb{P} ROLECONT \bullet invalidContracts = invalidroleContracts(agent?) \wedge invalidContracts \neq \emptyset \wedge$

$\exists uagent : AGENT \bullet uagent = updateAgent_{contracts}(agent?, agent?.roleContracts_a \setminus invalidContracts) \wedge$

$\exists affectedorgs : \mathbb{P} ORG \bullet affectedorgs = \{o : organizations \mid \exists rc : invalidContracts \bullet rc \in o.roleContracts_o\} \wedge$

$\exists uorgs : \mathbb{P} ORG \bullet uorgs = \{uo : ORG \mid \exists ao : affectedorgs \bullet$

$uo.name_o = ao.name_o \wedge uo.context_o = \{a : AGENT \mid a \text{ activeIn } uo \bullet a.context_a\} \wedge$

$uo.roleContracts_o = ao.roleContracts_o \setminus invalidContracts \wedge uo.rolePositions_o = ao.rolePositions_o\} \wedge$

$organizations' = organizations \setminus affectedorgs \cup uorgs \wedge$

$agents' = agents \setminus \{agent?\} \cup \{uagent\}$

AGENT

$name_a : AGENTNAME$

$context_a : AGENTCONTEXT$

$capabilities_a : \mathbb{P} CAPABILITY$

$roleContracts_a : \mathbb{P} ROLECONT$

$\forall rc : roleContracts_a \bullet rc.agentName = name_a$

ORG

$name_o : ORGNAME$

$context_o : ORGCONTEXT$

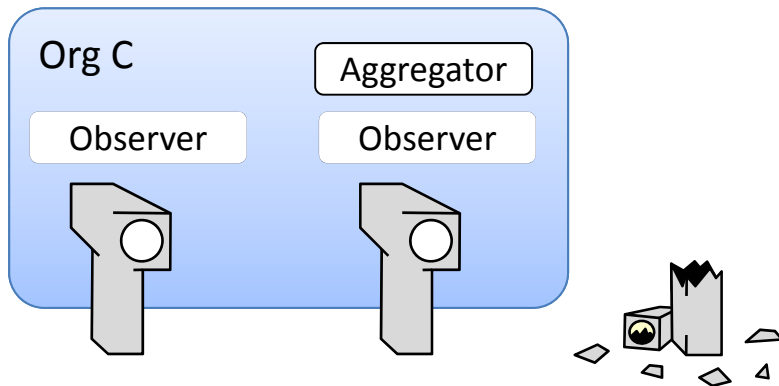
$roleContracts_o : \mathbb{P} ROLECONT$

$rolePositions_o : \mathbb{P} ROLEPOS$

$\forall rp : rolePositions_o \bullet rp.orgName = name_o$

$\forall rc : roleContracts_o \bullet$

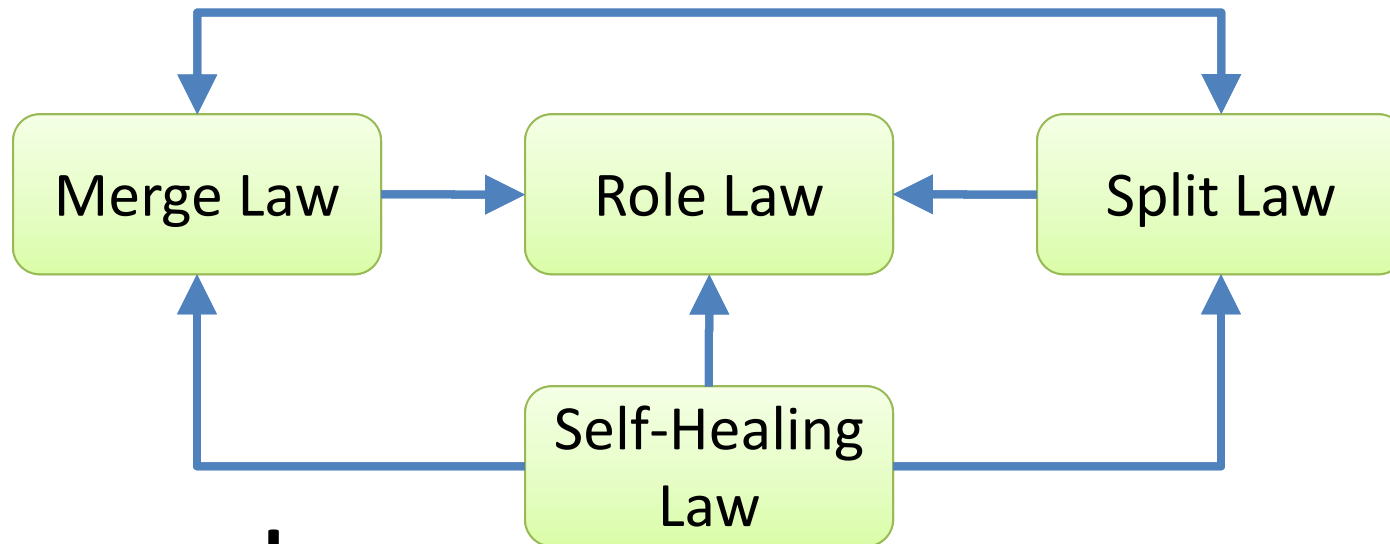
$rc.rolePosition.orgName = name_o$



# Summary

## Contribution: Rigorous description of

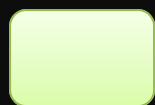
- Organizational concepts
- Middleware mechanisms



## Future work

- Models allow us to reason about properties and services offered by the middleware

**KEY**



Reflective Adaptation  
Process or Law



Trigger by Adapting  
Organization Model