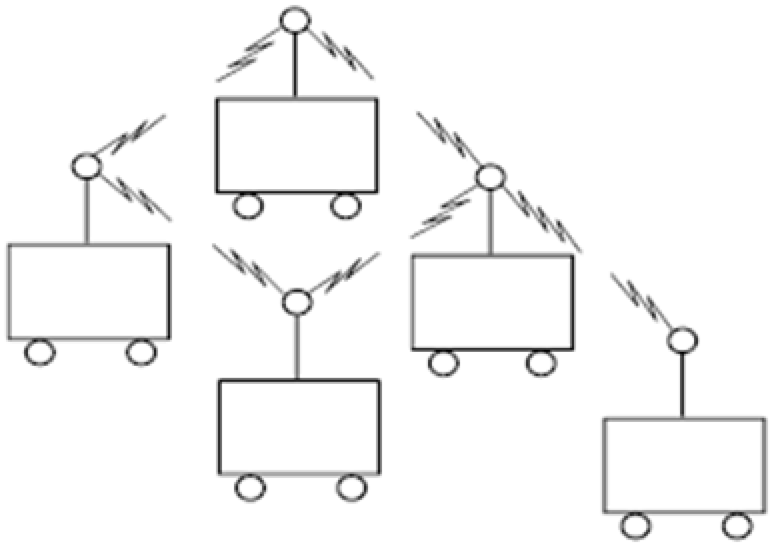
Creating Smart Agents:

A Distributed Control Theory for DES

Kai Cai
Osaka City University

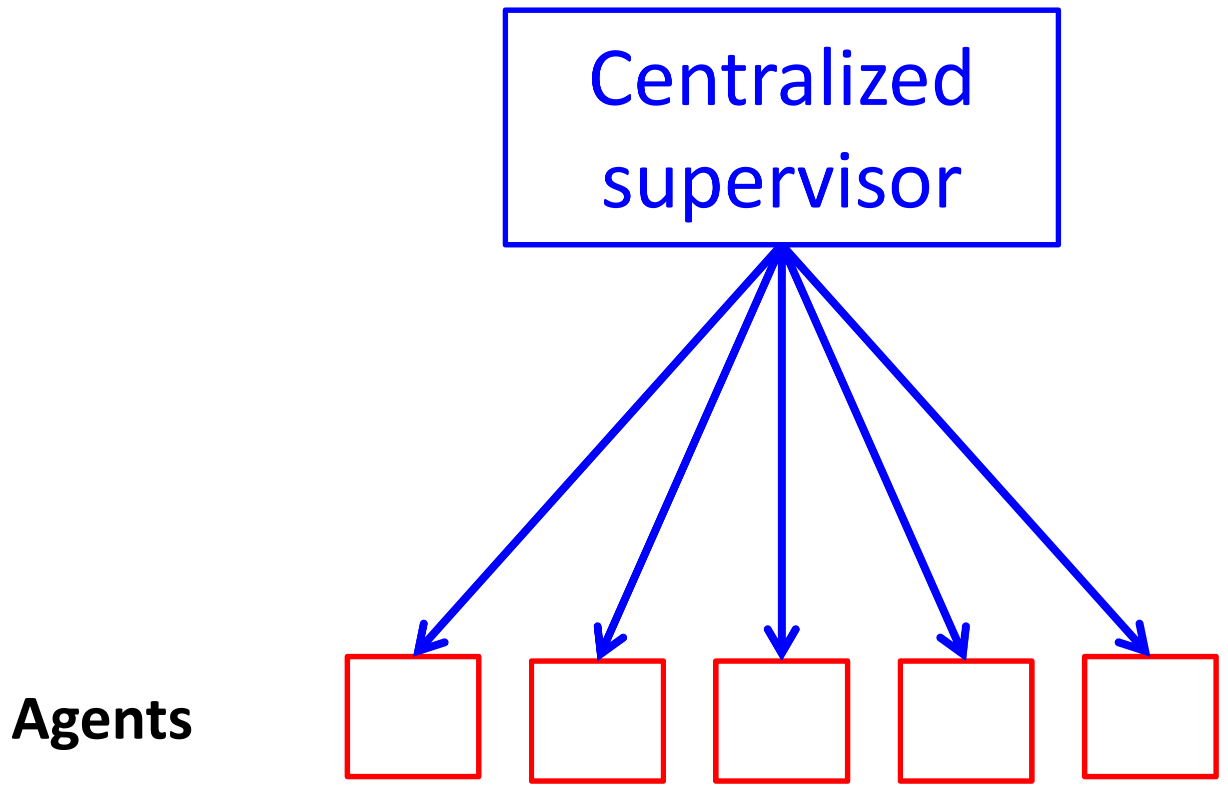
RW30 Workshop at CDC’17
2017.12.11

I am interested in multi-agent systems:

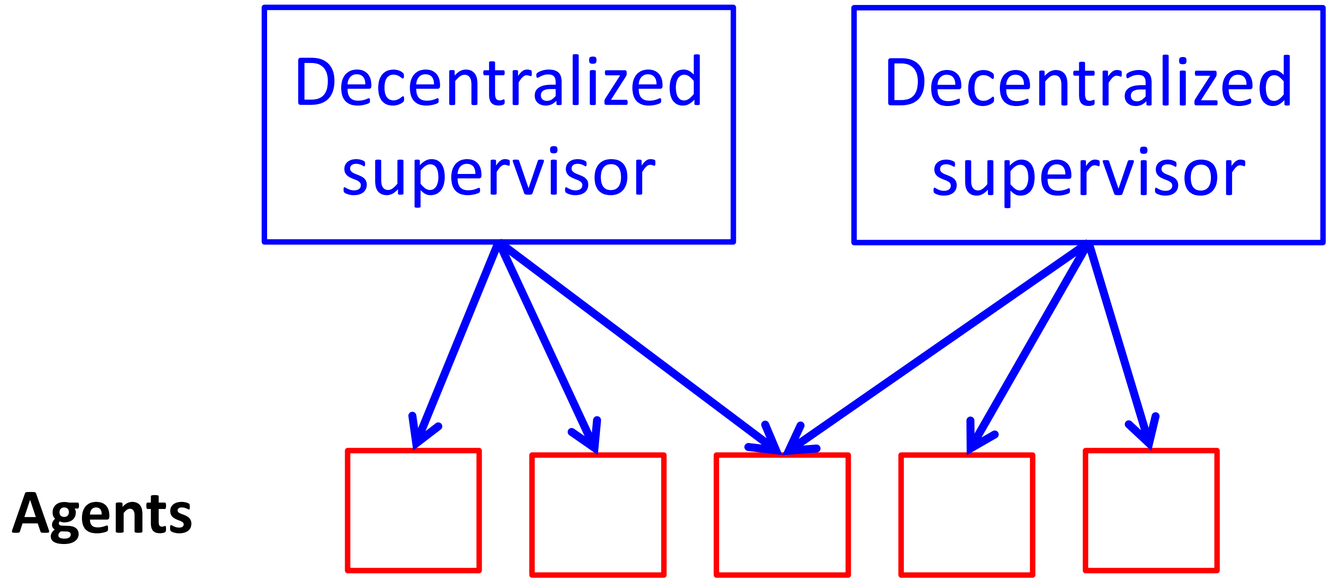


I am working on both agents with continuous dynamics and agents with discrete-event dynamics

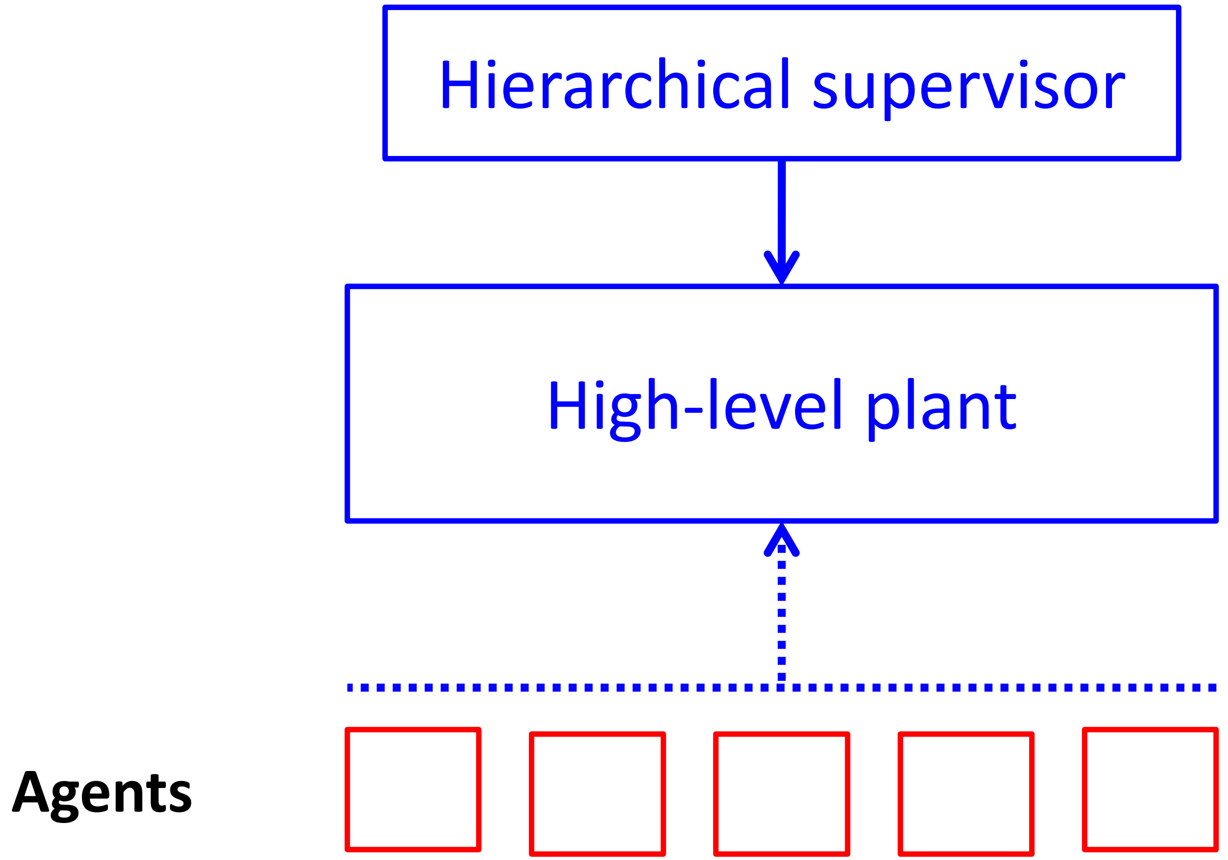
RW30: control architectural summary



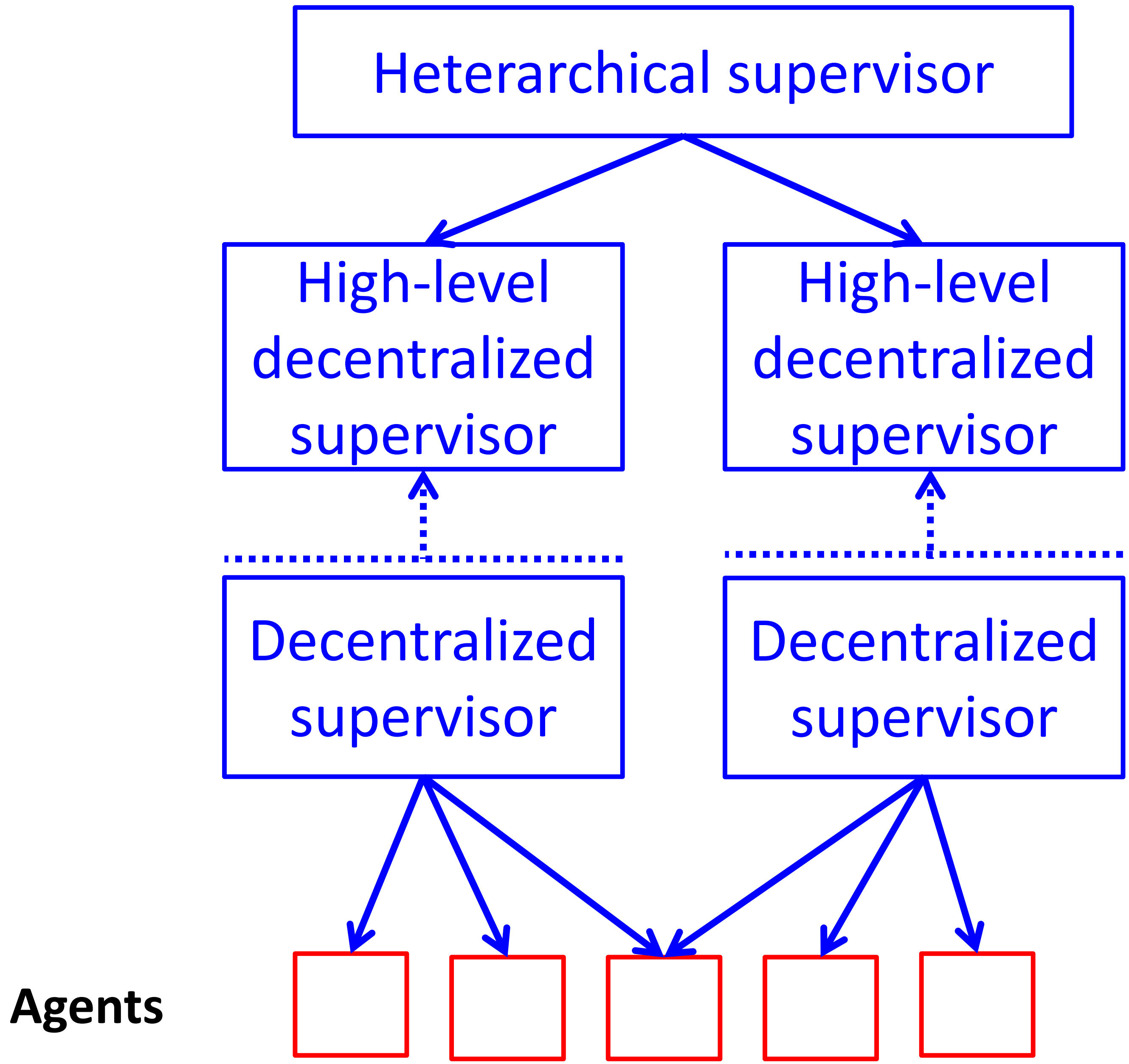
RW30: control architectural summary



RW30: control architectural summary



RW30: control architectural summary



RW30: control architectural summary

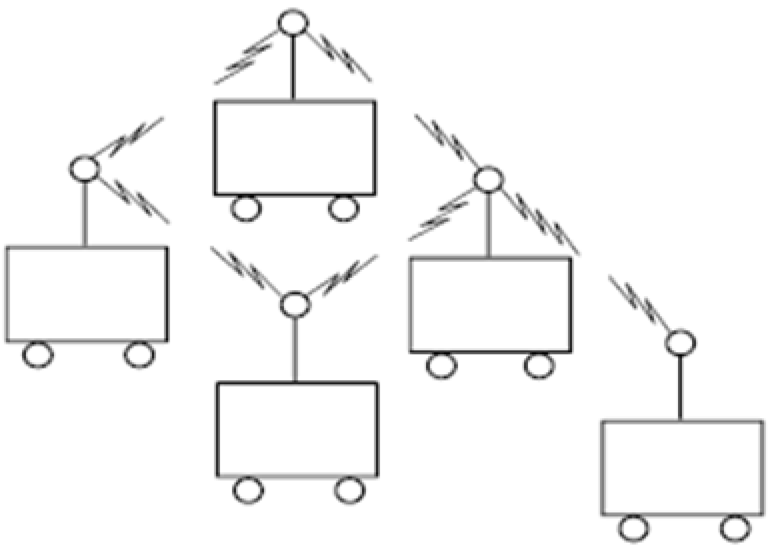
Centralized (monolithic): 1987-

Decentralized : 1988-

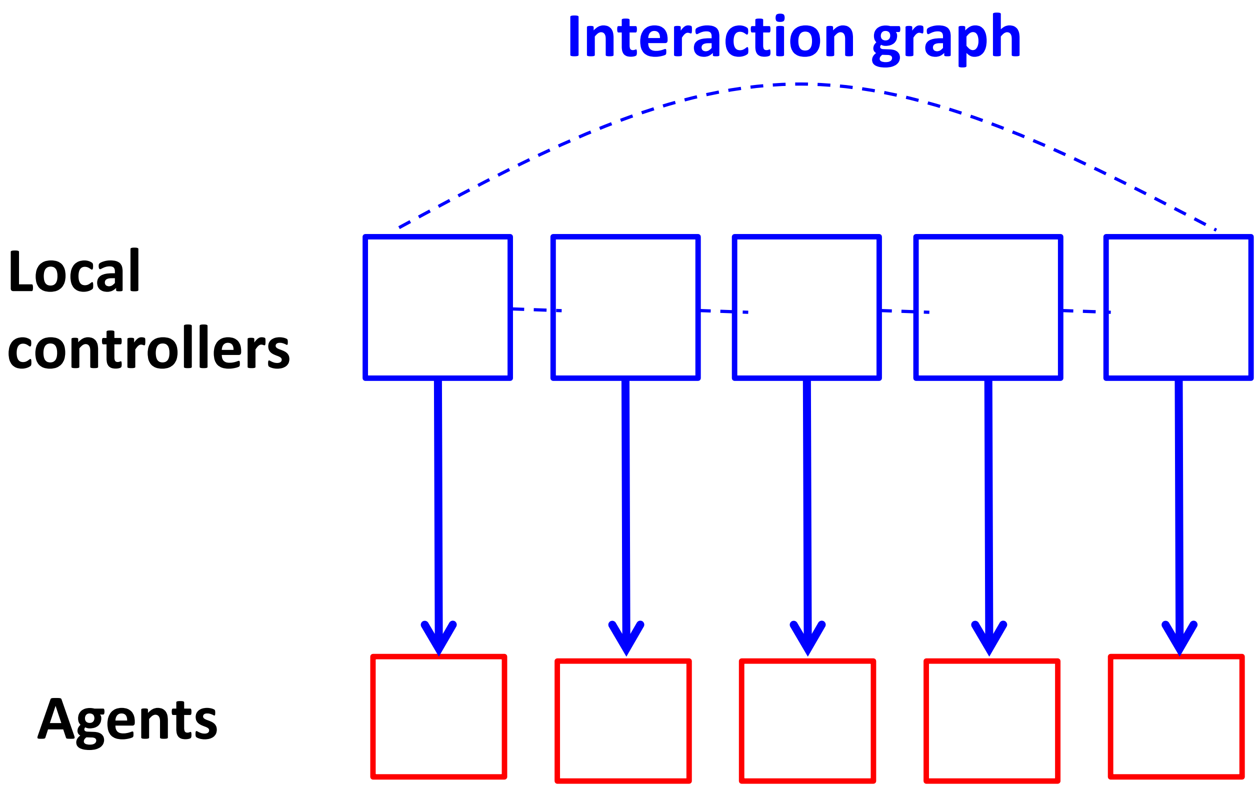
Hierarchical : 1991-

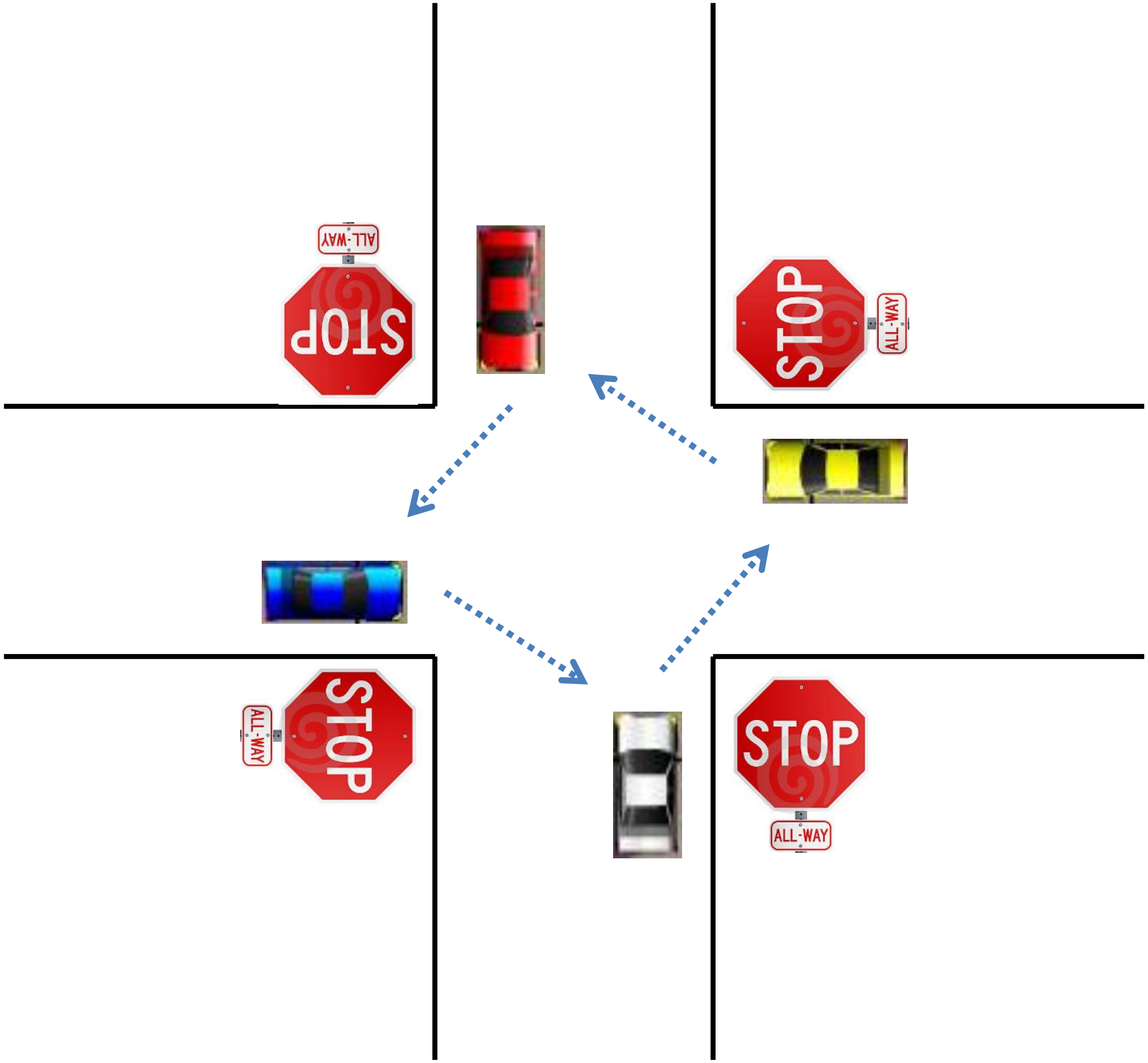
Heterarchical: 1996-

But, are these architectures suitable for controlling multi-agent systems, if individual agents are expected to be autonomous and smart



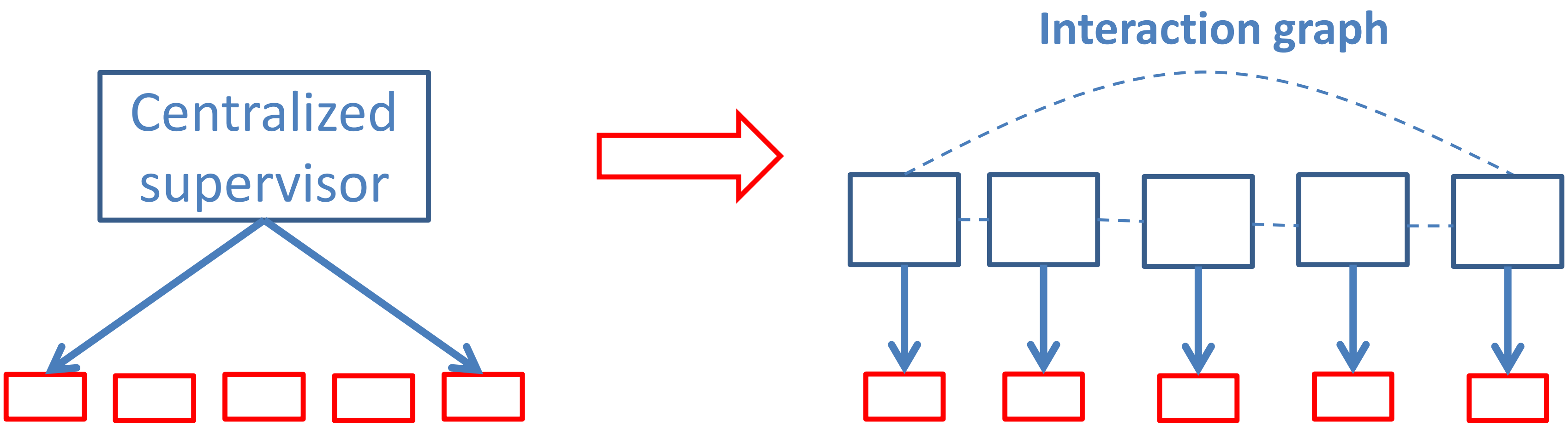
Distributed (for smart agents)




STOP
ALL-WAY

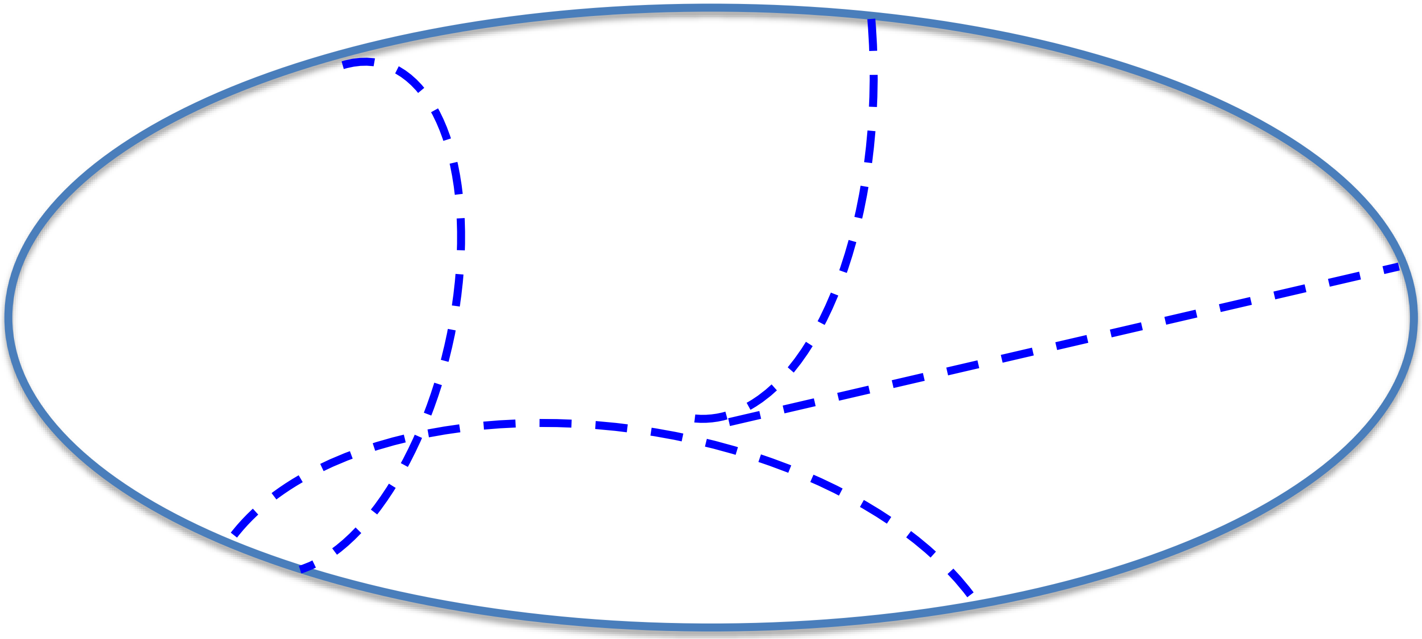
How to design distributed control?

Suppose we have: **Can we decompose the supervisor to get:**



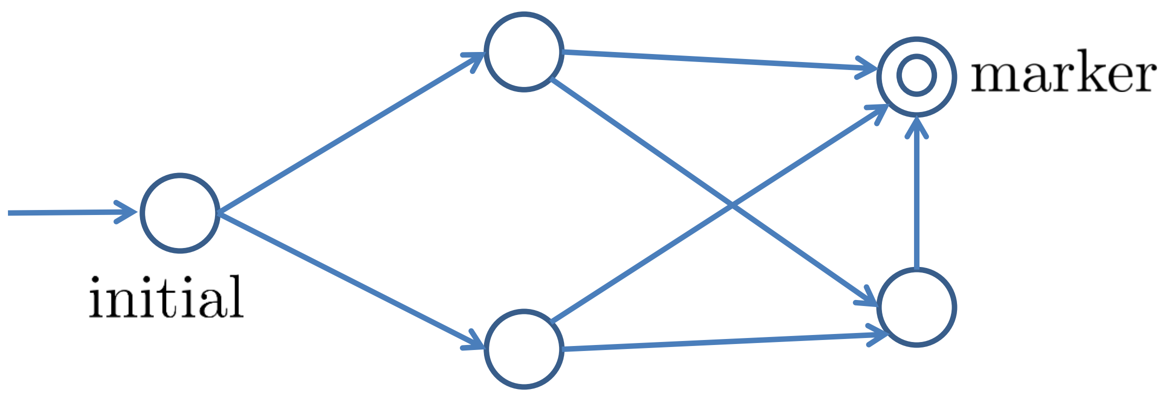
How to decompose?

SUP



control cover based on
agent i's control information

LOC$_i$



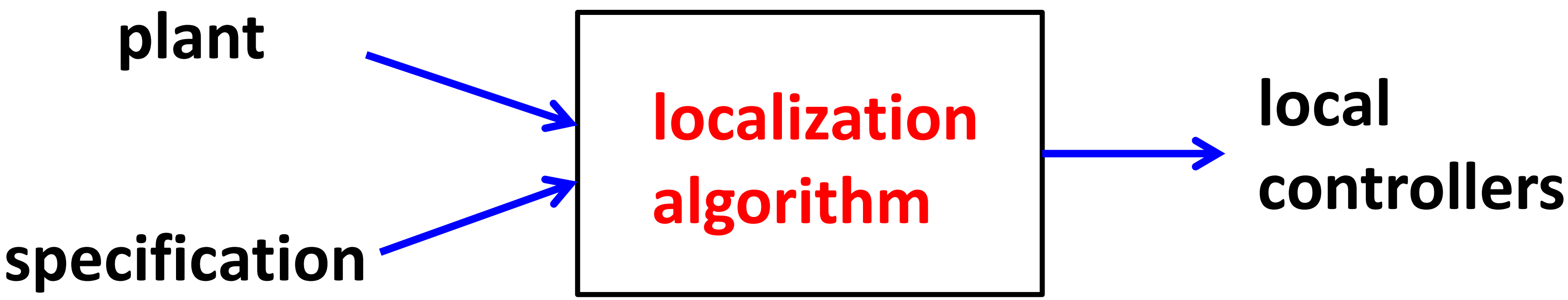
Local controllers are quotient dynamic systems *modulo* locally unavailable control information.

Supervisor localization for distributed control synthesis:

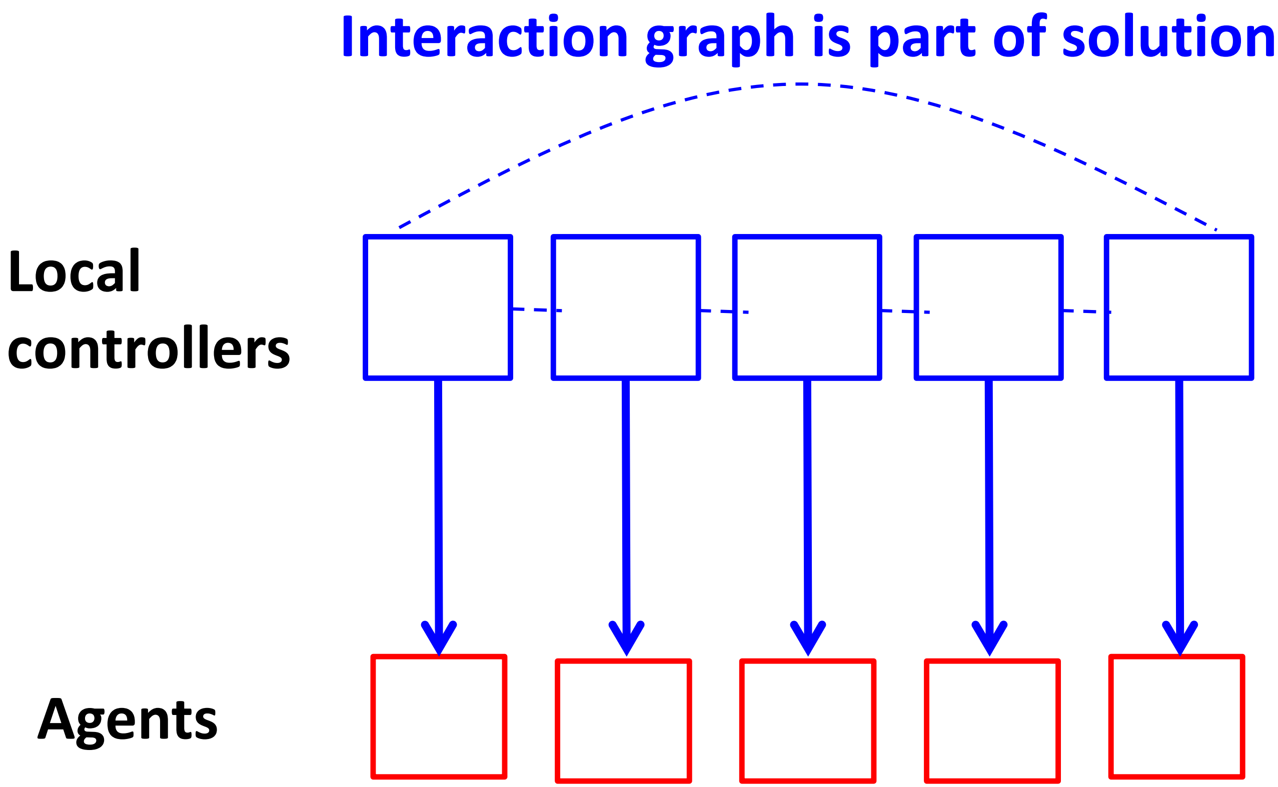
- Given N agents (plant)
- Given a collective behavior (specification)

- First, compute the centralized supervisor $\mathbf{SUP}$
  Then, decompose $\mathbf{SUP}$ into a set of
  local controllers $\mathbf{LOC}_i$ such that

$$\mathbf{LOC}_1 \;||\; \cdots \;||\; \mathbf{LOC}_N \text{ `='} \; \mathbf{SUP}$$


plant
specification
localization
algorithm
local
controllers

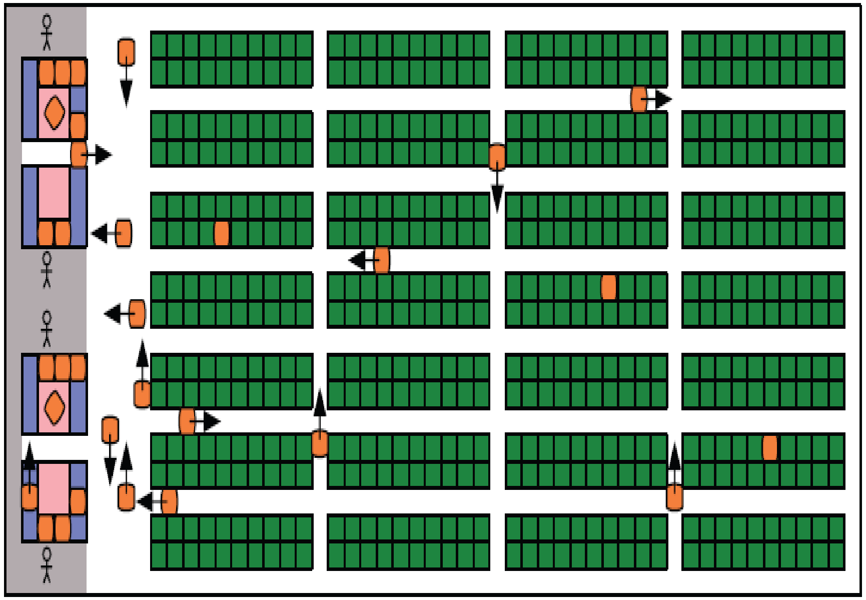
Distributed architecture emerges:

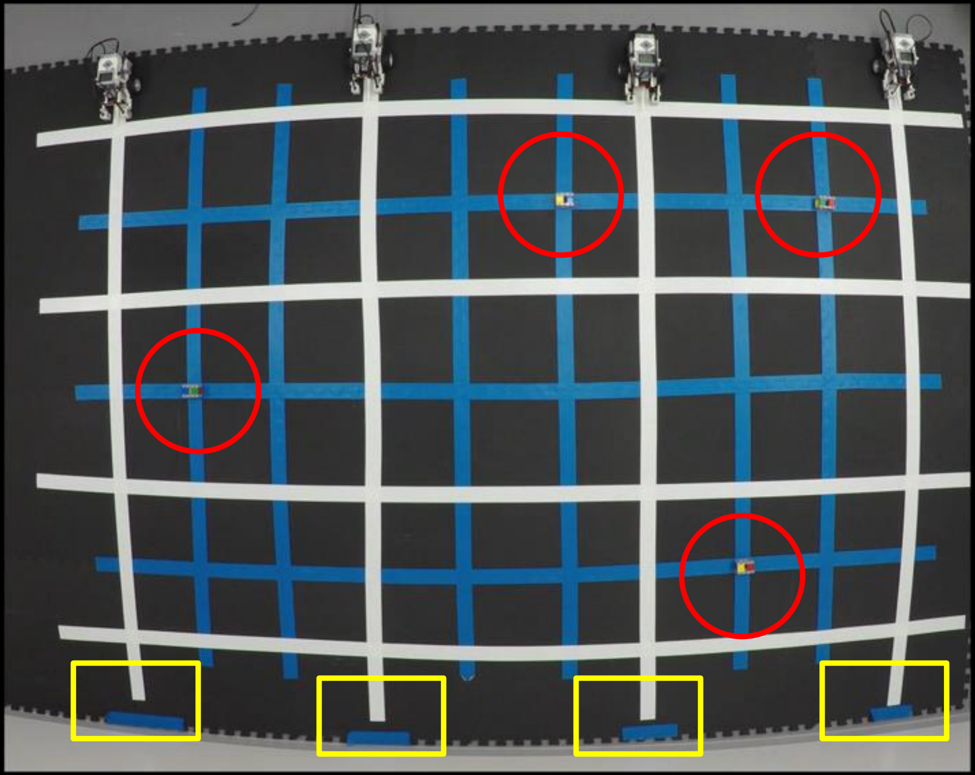


Example: warehouse automation

Kiva Systems

No collisions
No deadlocks

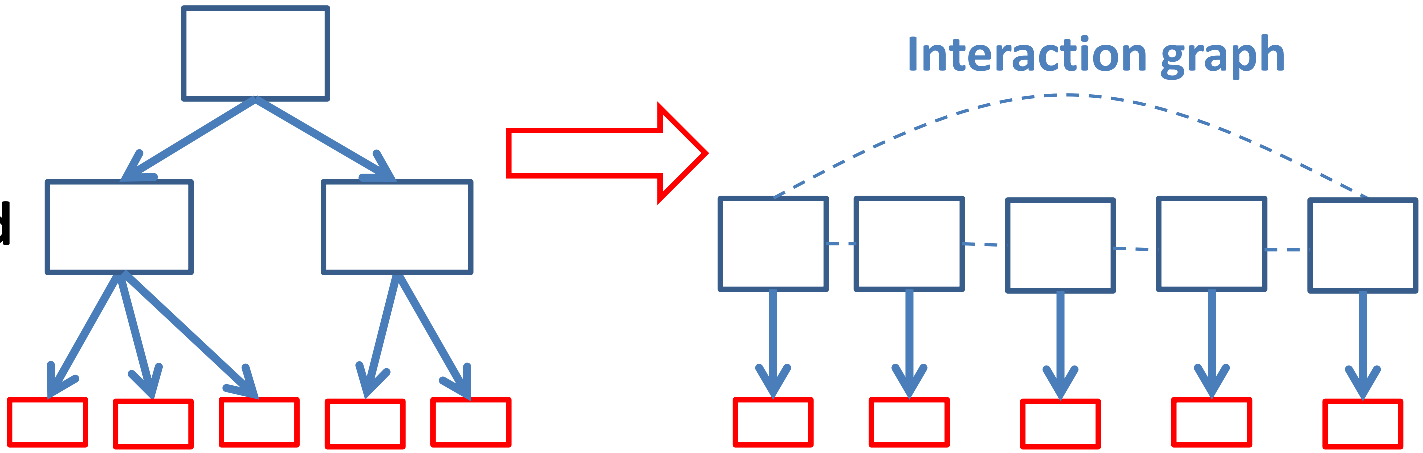




For large-scale systems

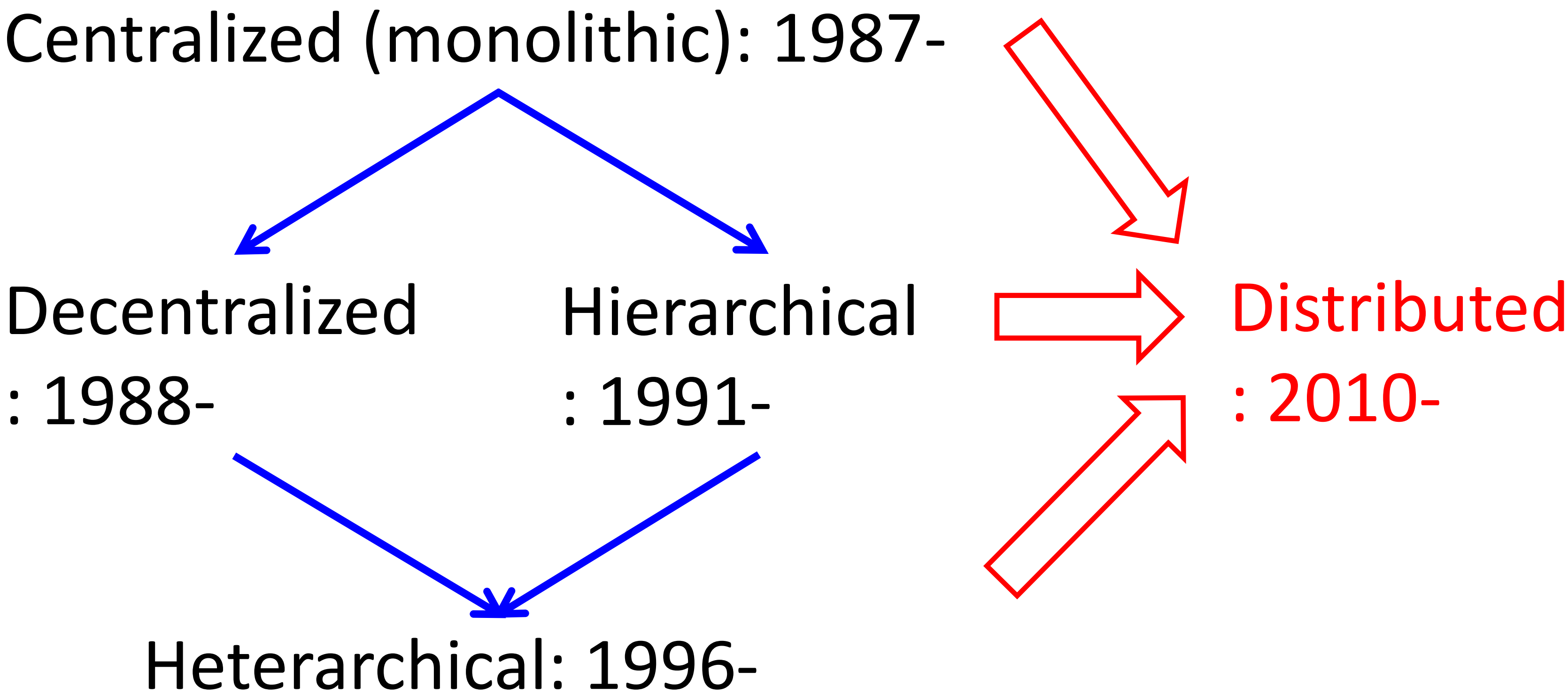
Hierarchical supervisor

Decentralized supervisors



Alternative supervisor localization

RW30: control architectural summary



Localization for timed DES

Standard RW framework: logical specifications

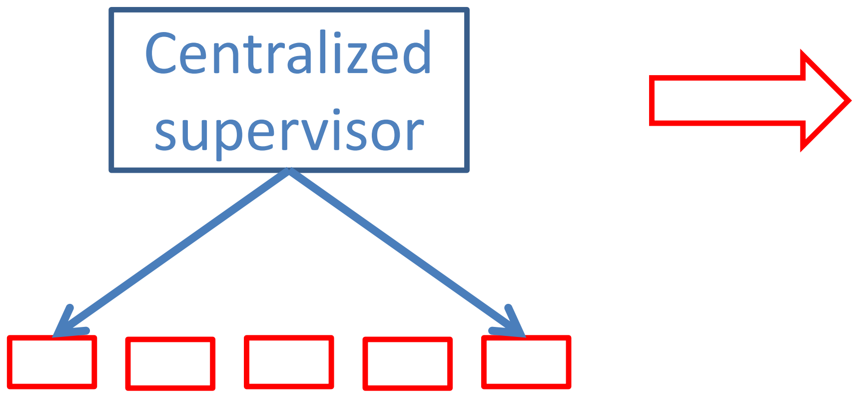
RW framework + time: temporal specifications

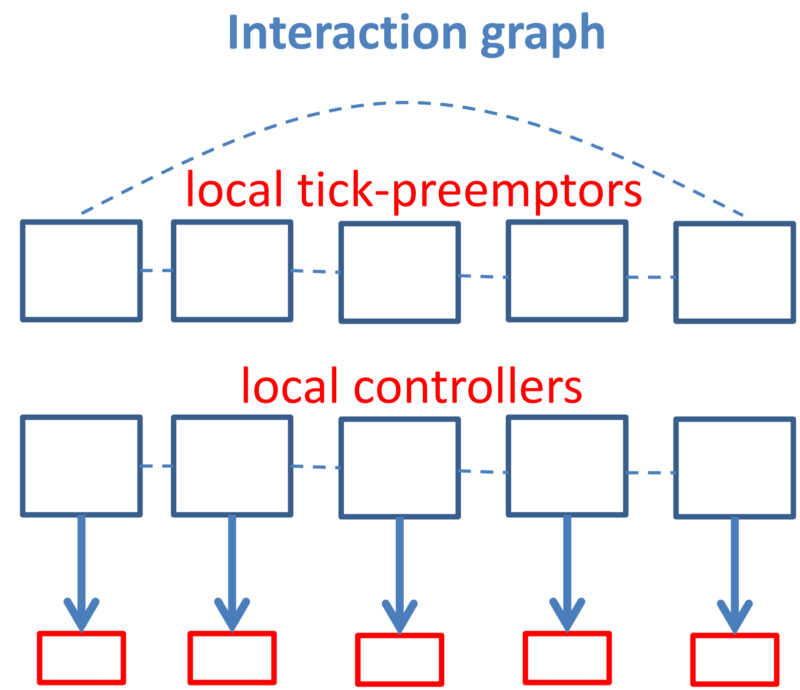
tick event: time delay, hard deadline

Localization for timed DES

Enforcing:

- temporal specifications
- logical specifications





Separation Principle

Localization for infinite-behavior DES

RW framework + time:
finite-behavior temporal specifications

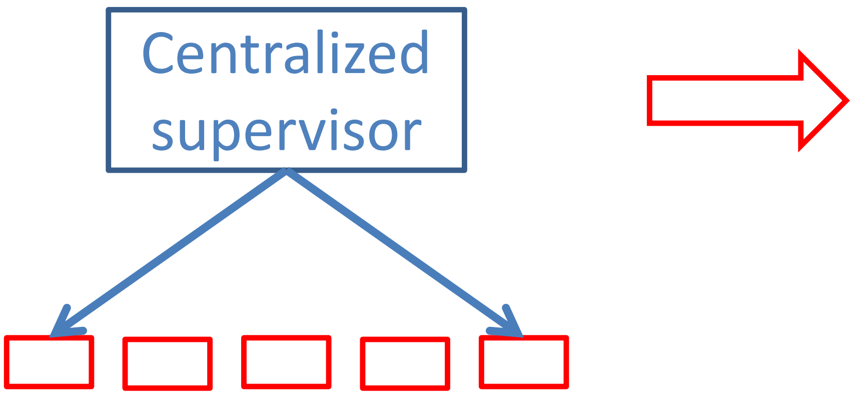
RW framework + temporal logic:
infinite-behavior temporal (liveness) specifications

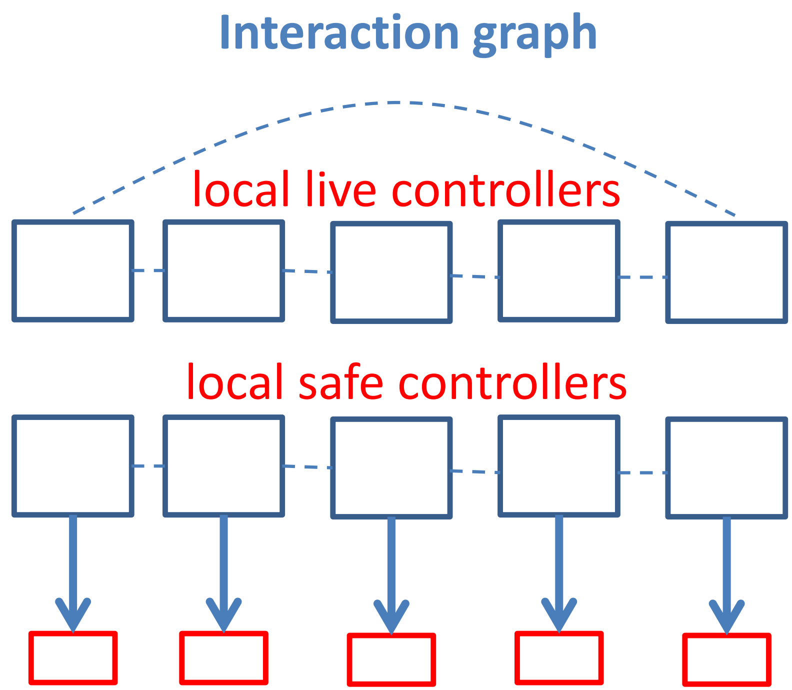
ω-languages: eventually, infinitely often, always,

Localization for infinite-behavior DES

Enforcing:

- infinite specifications
- finite specifications





Separation Principle

Conclusions

- Supervisor localization as a distributed control theory for discrete-event systems
  - Anything you can do globally, you can do locally
- Localization theory
  - Controllability
  - Observability
  - Opacity / diagnosability / detectability
  - State-minimization, Bisimulation-minimization

Localization: limitations and promises

Top-down:

Global control design

Local implementation

Bottom-up:

Global verification

Local control design

Convergence: **top-down inspired bottom-up**

W R

Workshop CDC'14, Los Angels