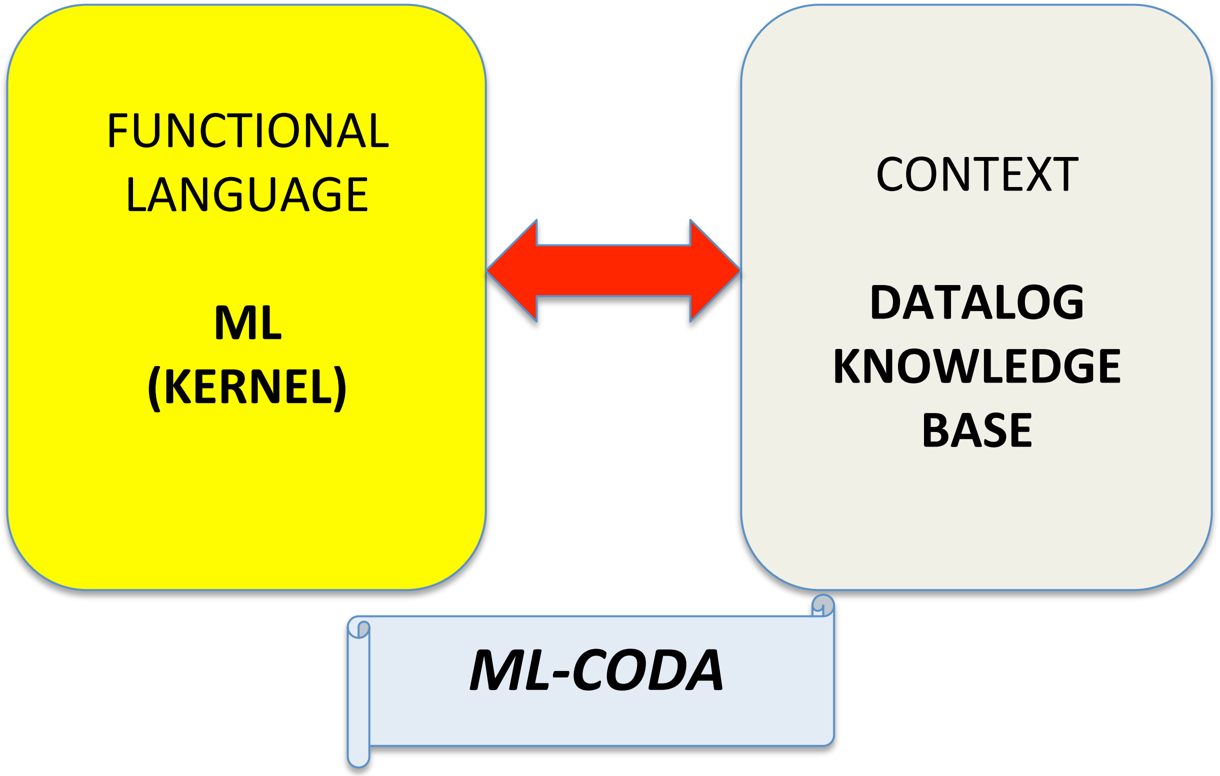
CONTEXT-ORIENTED PROGRAMMING ABSTRACTIONS FOR FOG

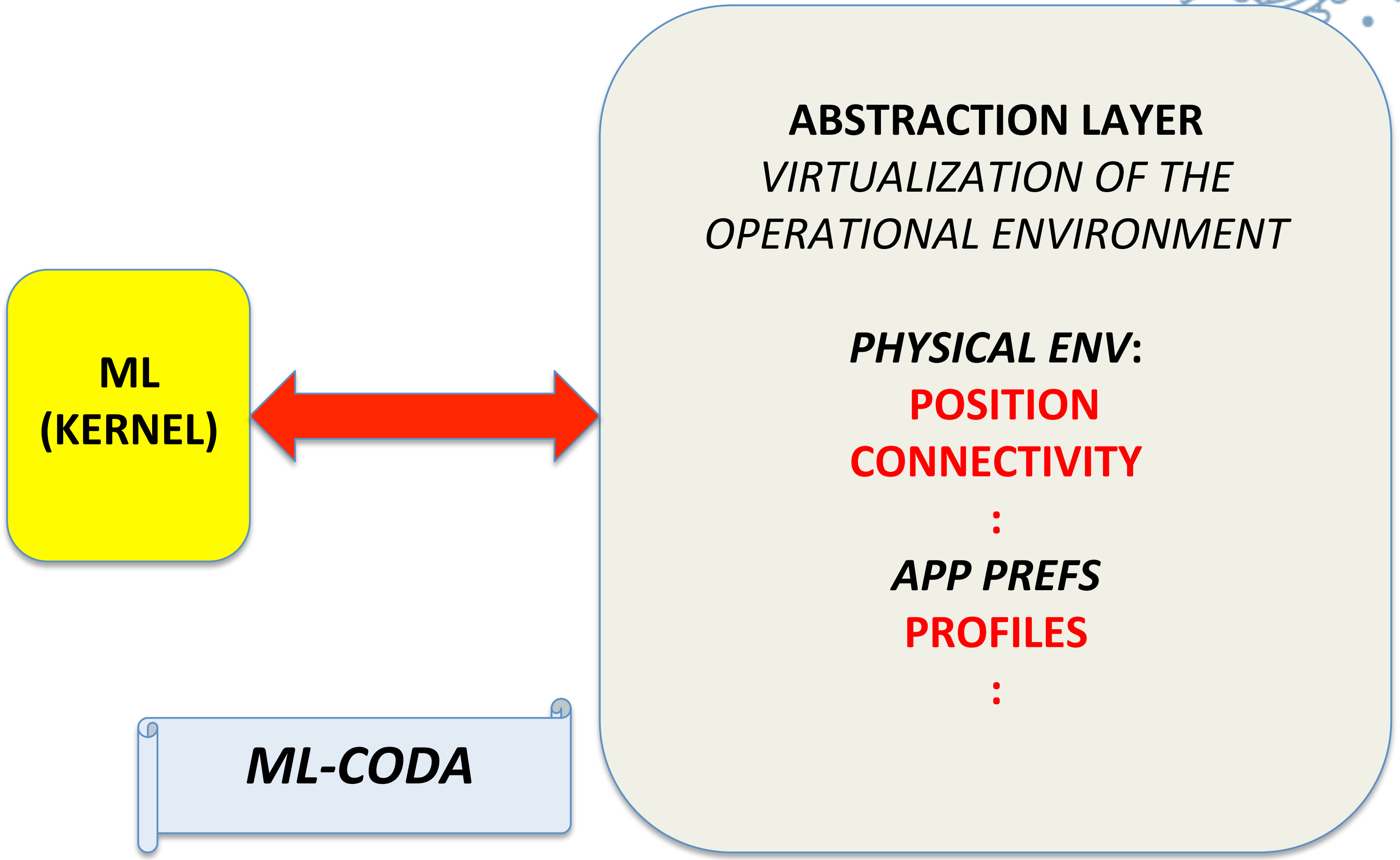
C. BODEI, P. DEGANO, G. FERRARI, L. GALLETTA

ML-CODA: A Context-Oriented Programming Language

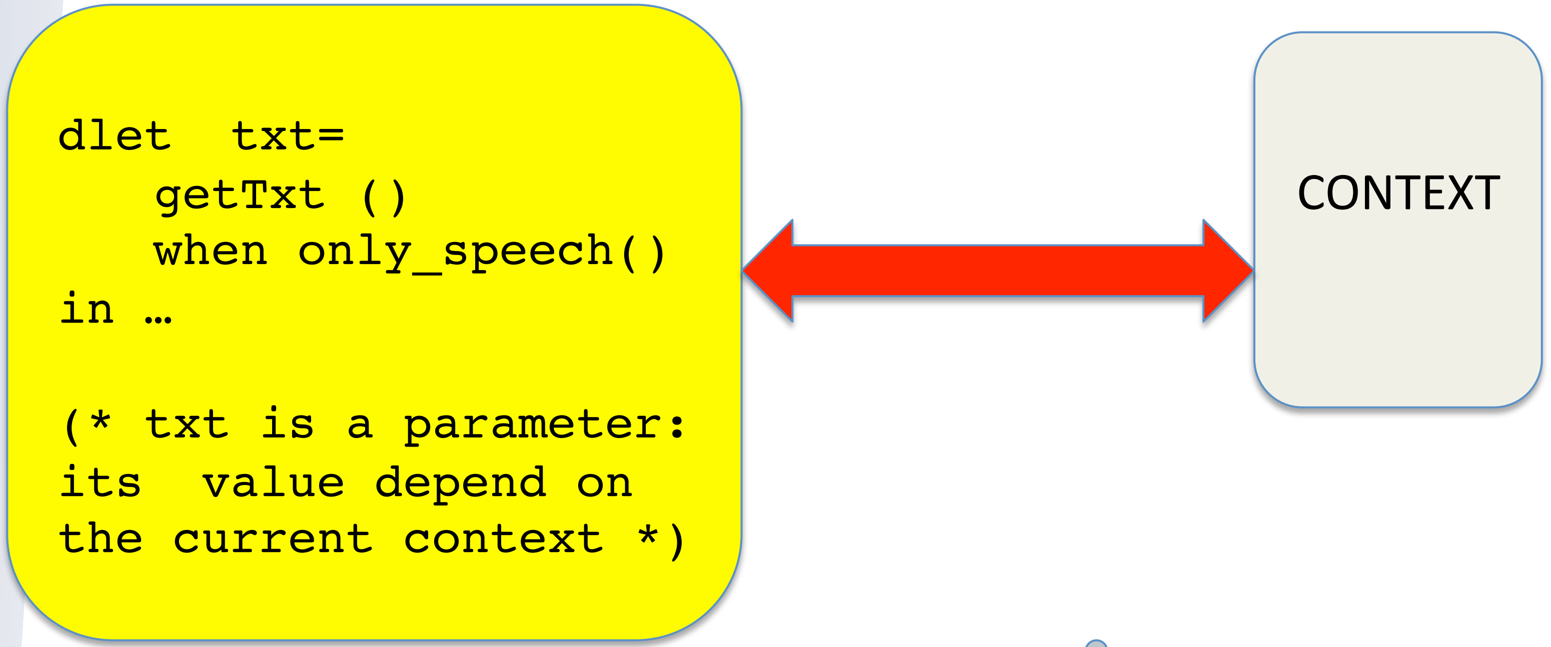


BODEI, CANCIANI, DEGANO, GALLETTA, SALVATORI, FERRARI

CONTEXT


ML
(KERNEL)
ABSTRACTION LAYER
VIRTUALIZATION OF THE OPERATIONAL ENVIRONMENT
***PHYSICAL ENV*:**
POSITION
CONNECTIVITY
:
APP PREFS
PROFILES
:
ML-CODA

Context Dependent Binding

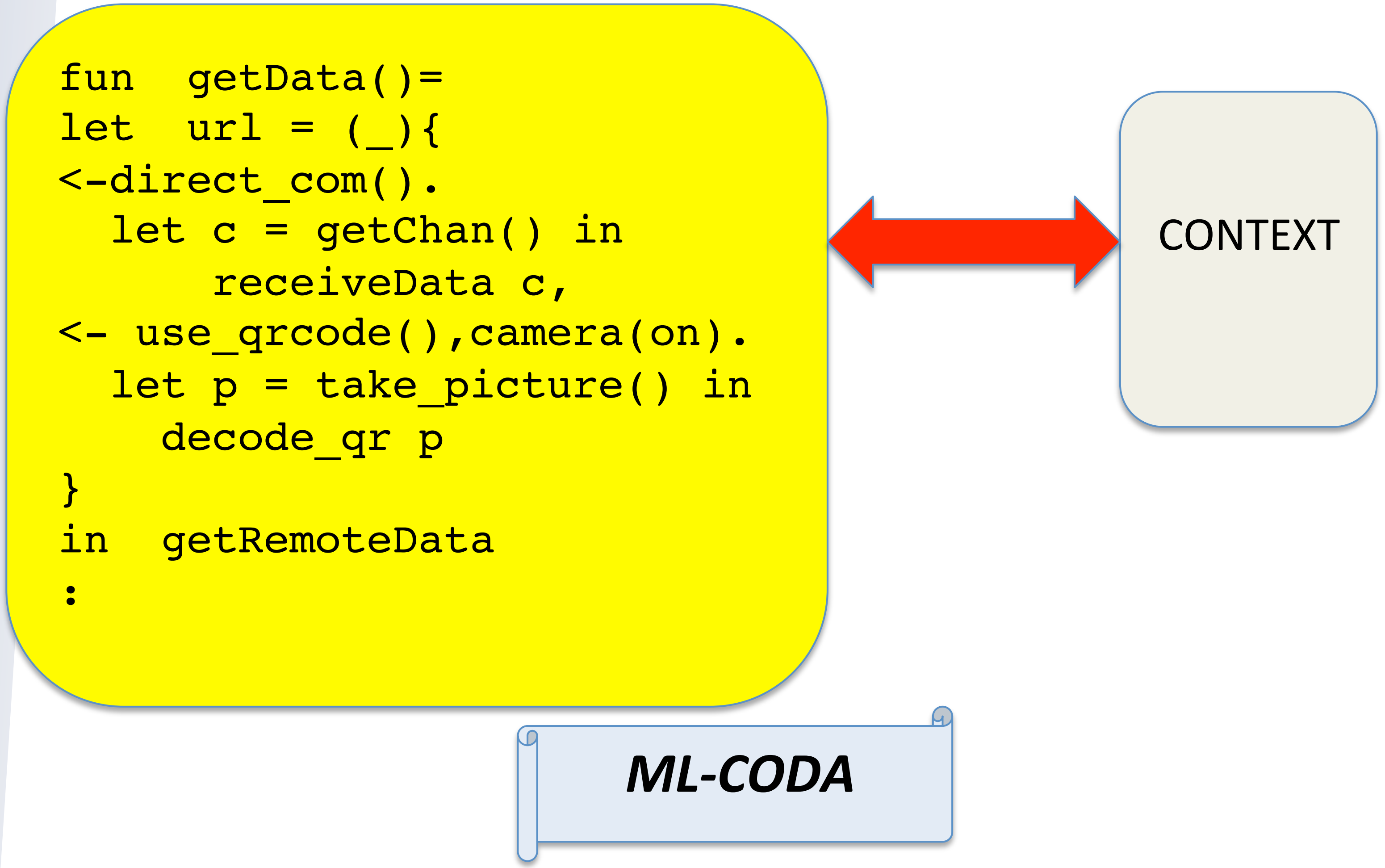


```
dlet  txt=
    getTxt ()
    when only_speech()
in …

(* txt is a parameter:
its  value depend on
the current context *)
```

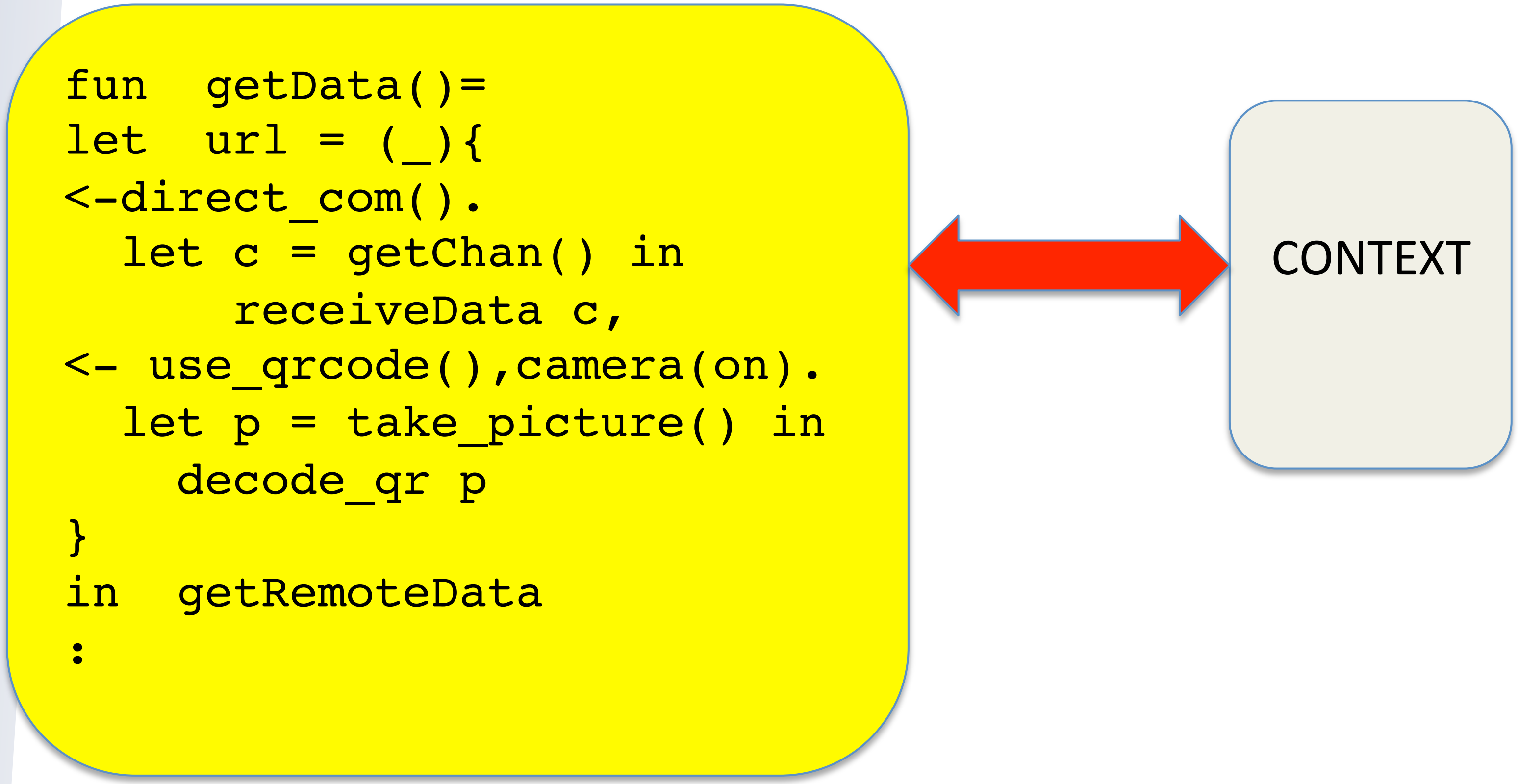
ML-CODA

Behavioural Variations



```
fun  getData()=
let  url = (_){
<-direct_com().
  let c = getChan() in
      receiveData c,
<- use_qrcode(),camera(on).
  let p = take_picture() in
    decode_qr p
}
in  getRemoteData
:
```

Behavioral Variations



```
fun  getData()=
let  url = (_){
<-direct_com().
  let c = getChan() in
      receiveData c,
<- use_qrcode(),camera(on).
  let p = take_picture() in
    decode_qr p
}
in  getRemoteData
:
```

ML-CODA

Adaptivity: app can modify its behaviour according to changes in its context

ML-CODA

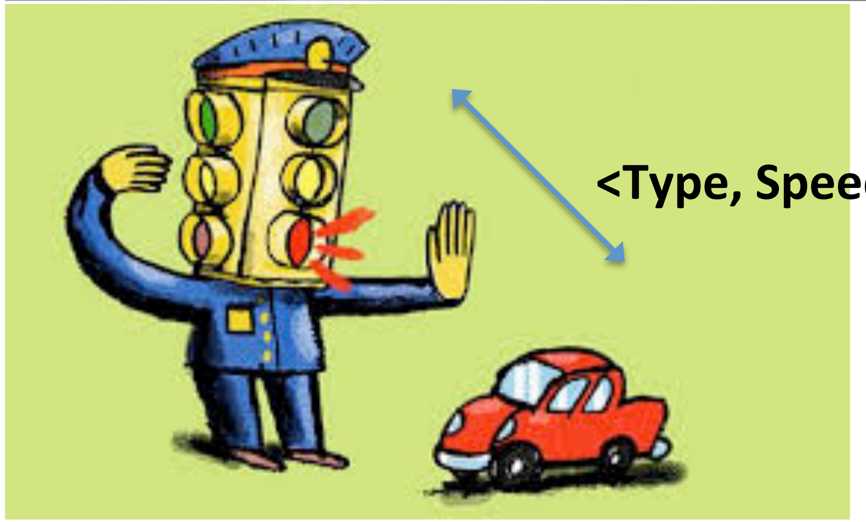
- Static Machinery (DFG@IEEE-TSE)
  - verify that dispatching mechanism always succeed
- Security Analysis (DBGS@JCS)
  - detect potential unsafe modications
- Prototype Implementation (CDFG@FOCLASA)
  - Context Oriented Extension of F#

OUR F(r)OG GOAL: PROGRAMMING ABSTRACTIONS in a Context-Aware fashion

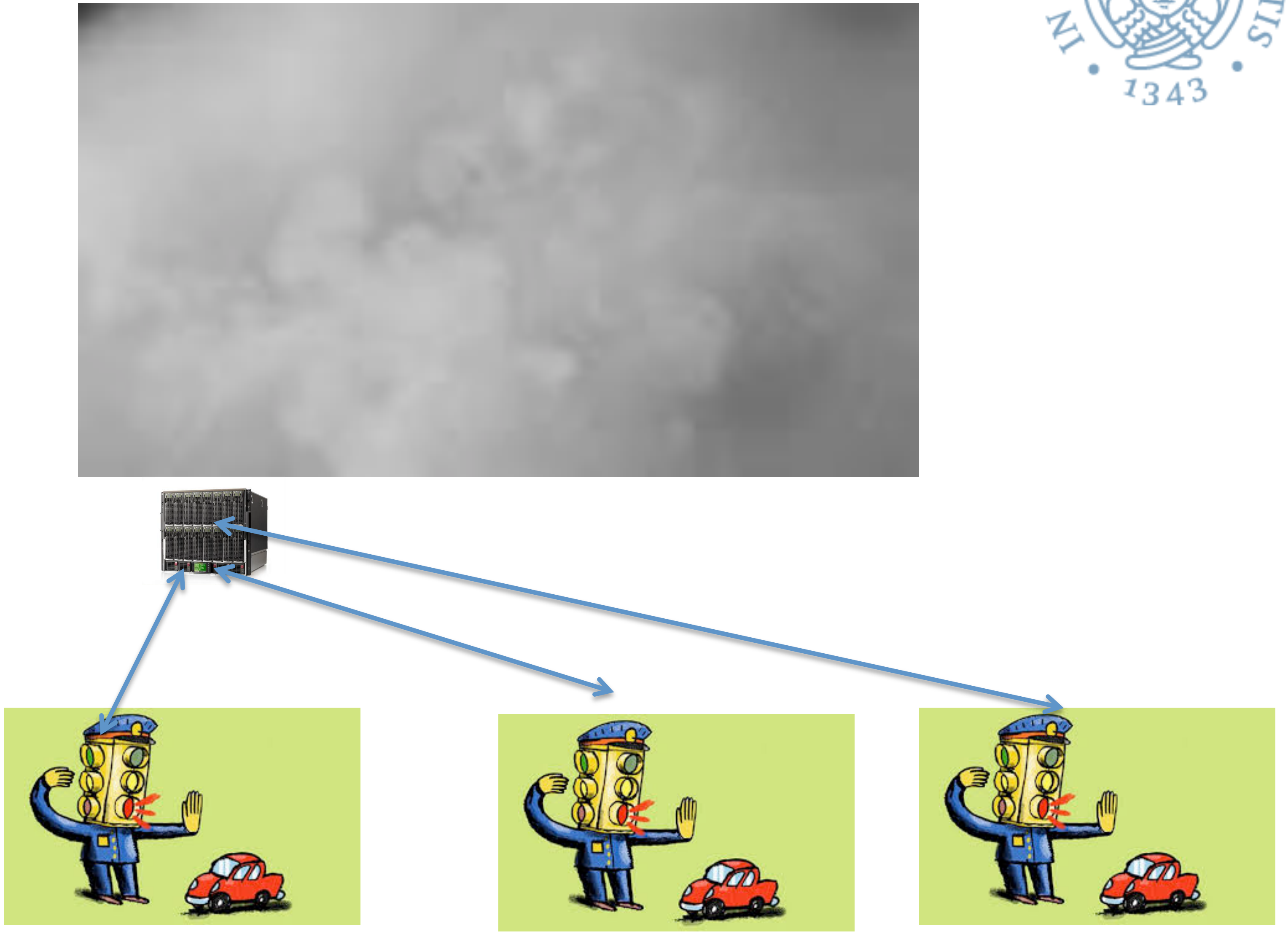
Programming Model

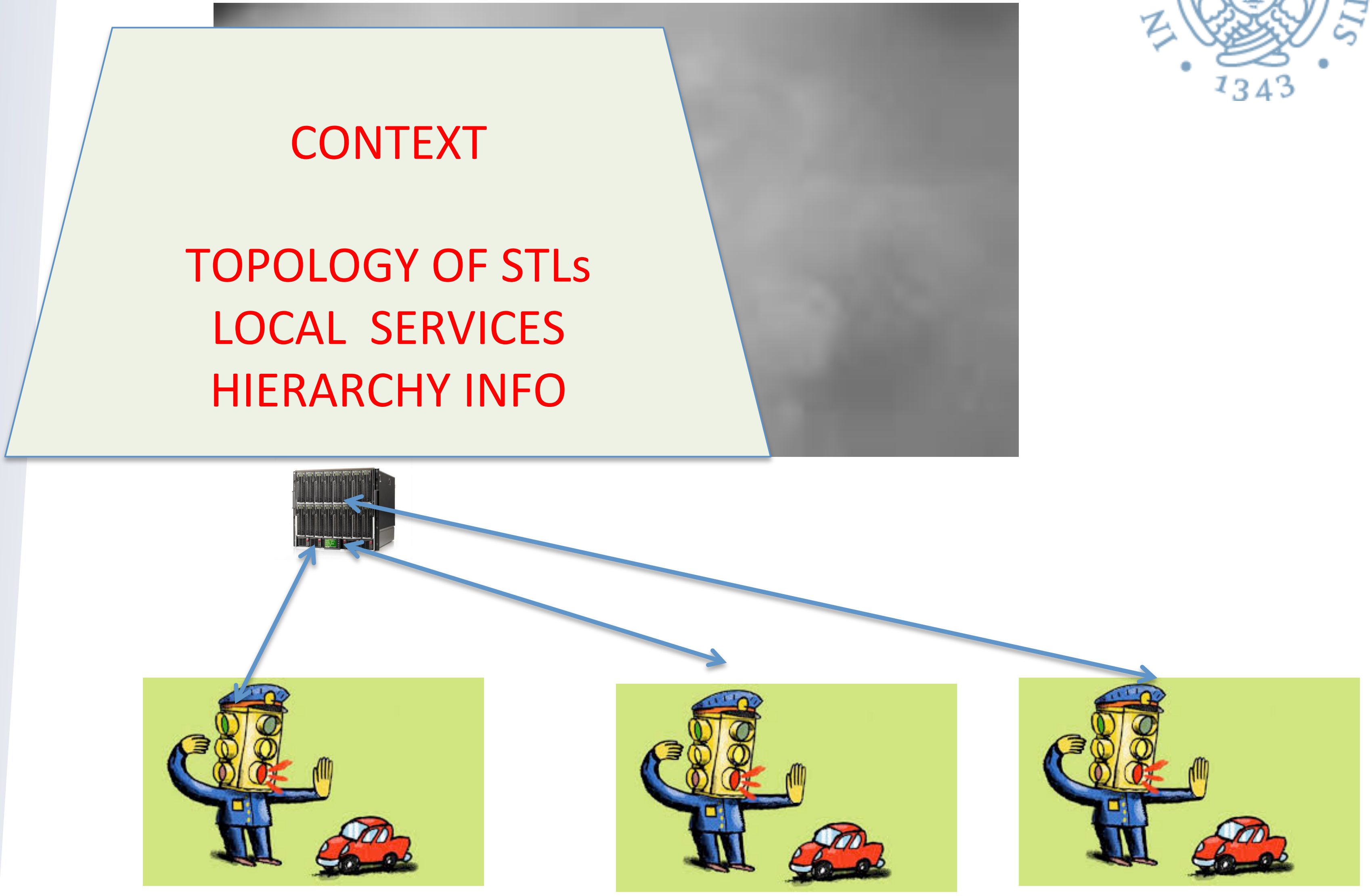
- How can we easily develop applications on the fog computing infrastructure?
  - Mirko's talk for further motivations
- Need a right programming model that
  - **Provides suitable programming abstractions**
  - **Ensures dynamic adaptation**
  - **Support context-aware orchestrations**
  - **Supports hierarchical resources**
  - **Enforce context-aware security properties**
  - **Support verification**

Our F(r)ROG goals: by examples



<Type, Speed, other info>



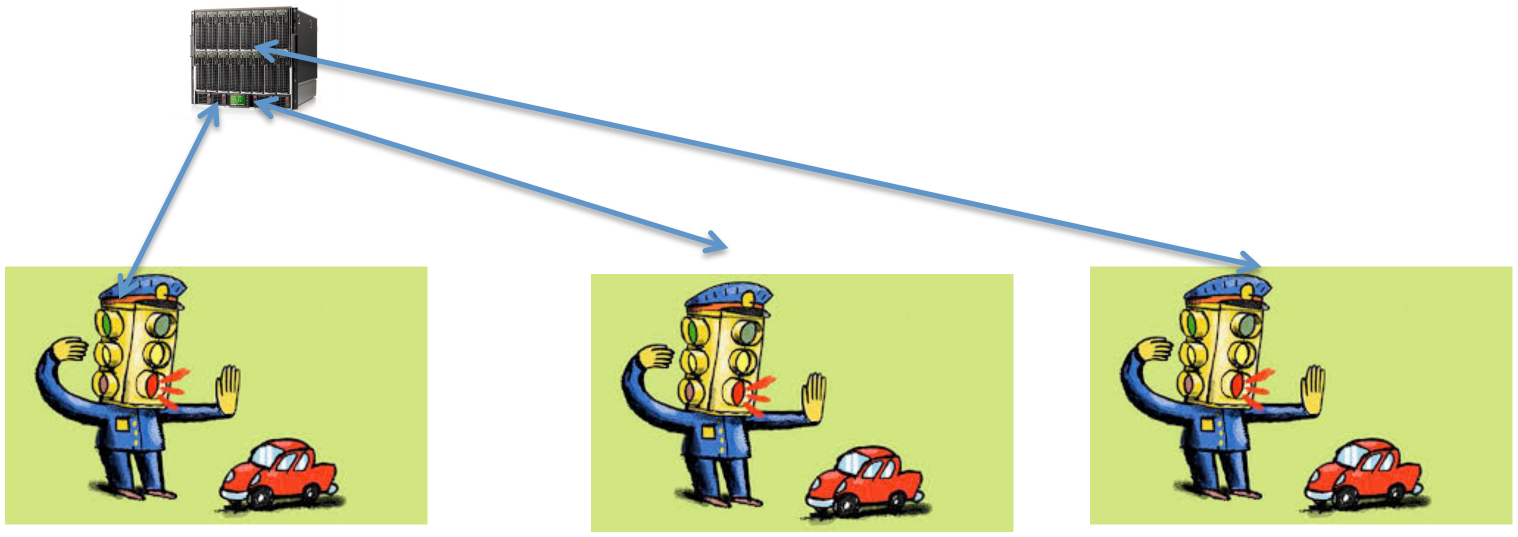

CONTEXT
TOPOLOGY OF STLs
LOCAL SERVICES
HIERARCHY INFO

COMPUTING & ORCHESTRATION

PUT_DATA()
LOCAL & NON LOCAL

CONTEXT

TOPOLOGY OF STLs
LOCAL SERVICES
HIERARCHY INFO



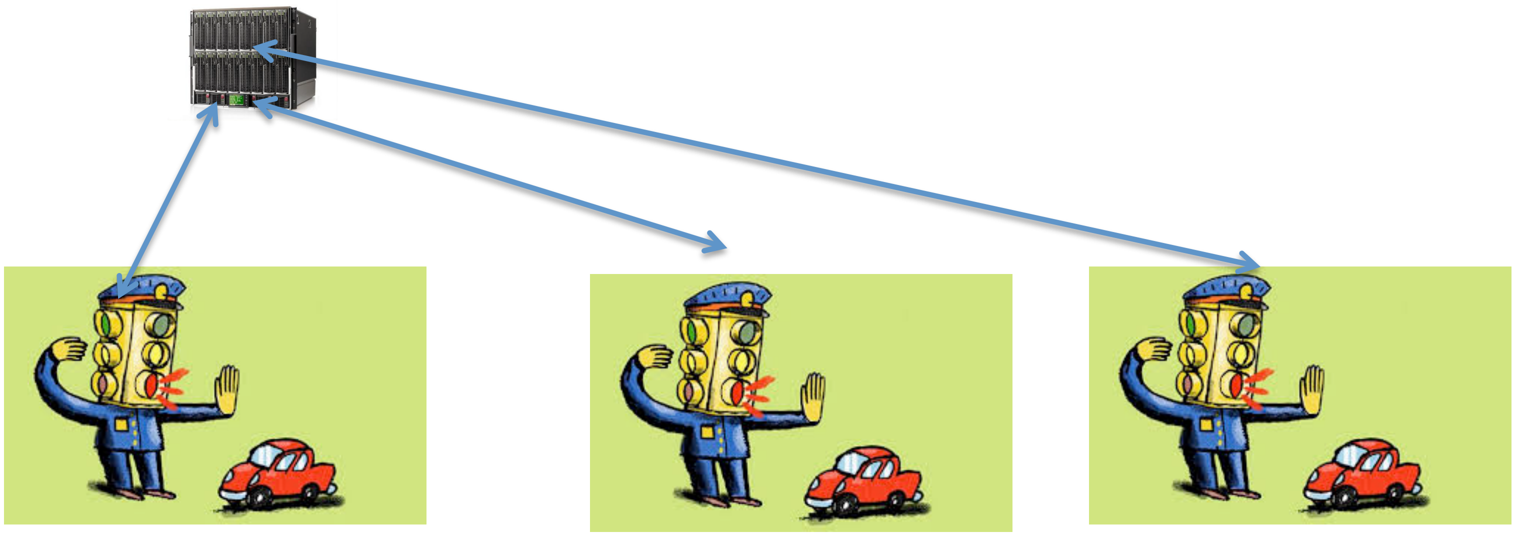
COMPUTING &
ORCHESTRATION

RCHESTRATE(param)

Slow-down warning

CONTEXT

TOPOLOGY OF STLs
LOCAL SERVICES
HIERARCHY INFO



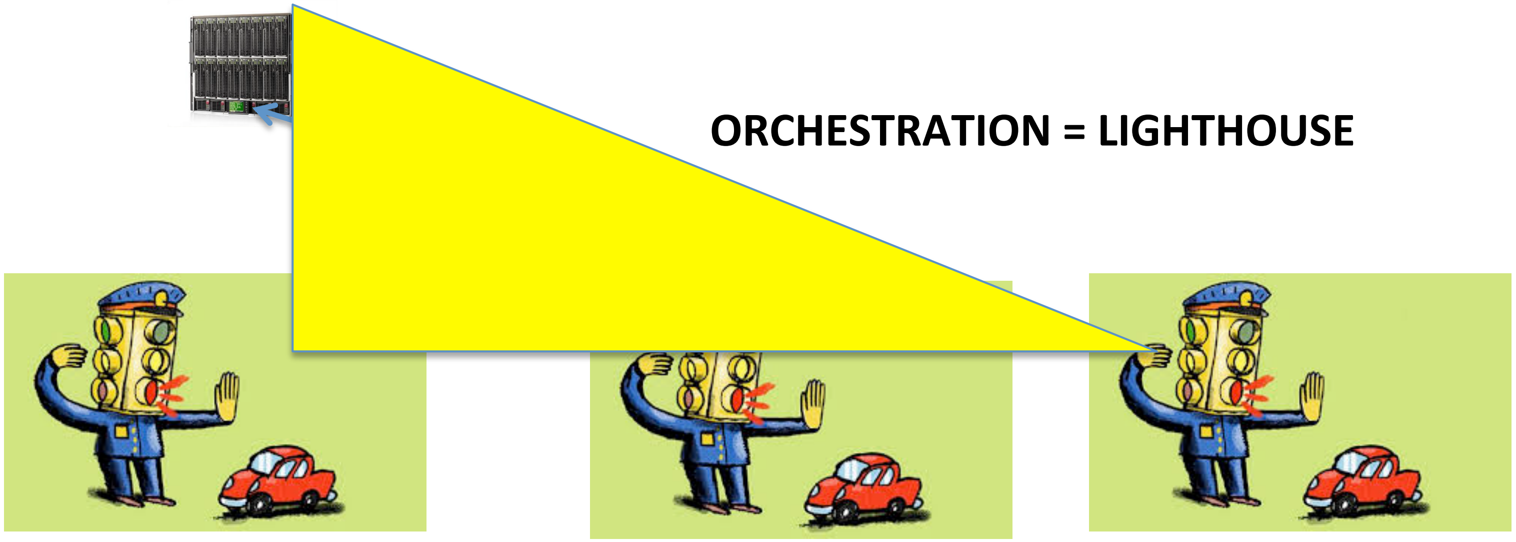
COMPUTING &
ORCHESTRATION

ORCHESTRATE()
Slow-down warning

CONTEXT

TOPOLOGY OF STLs
LOCAL SERVICES
HIERARCHY INFO

ORCHESTRATION = LIGHTHOUSE

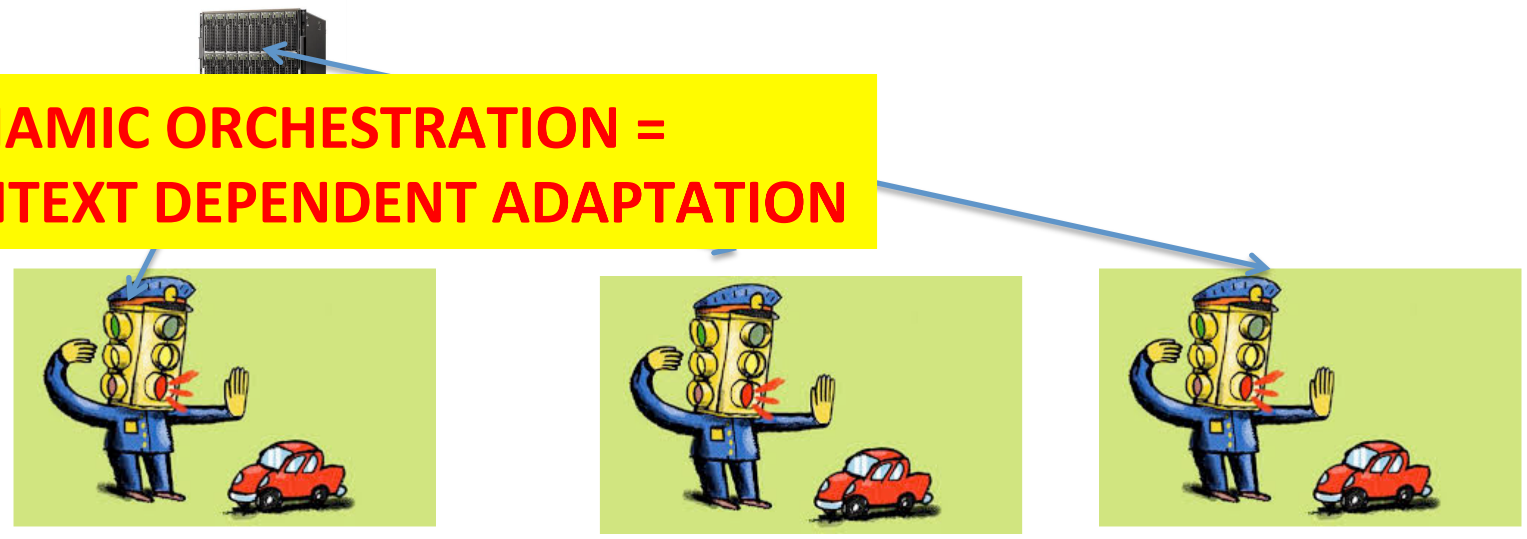


COMPUTING &
ORCHESTRATION

ORCHESTRATE()
Slow-down warning

CONTEXT

TOPOLOGY OF STLs
LOCAL SERVICES
HIERARCHY INFO



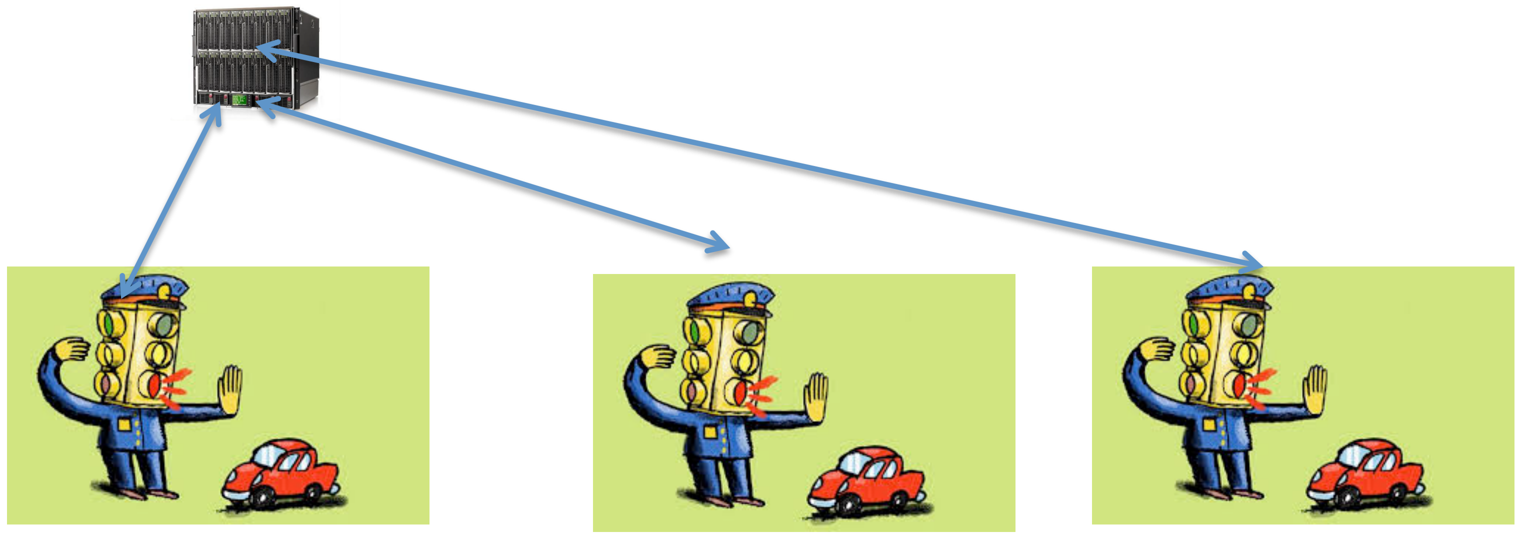
**DYNAMIC ORCHESTRATION =
CONTEXT DEPENDENT ADAPTATION**

COMPUTING & ORCHESTRATION

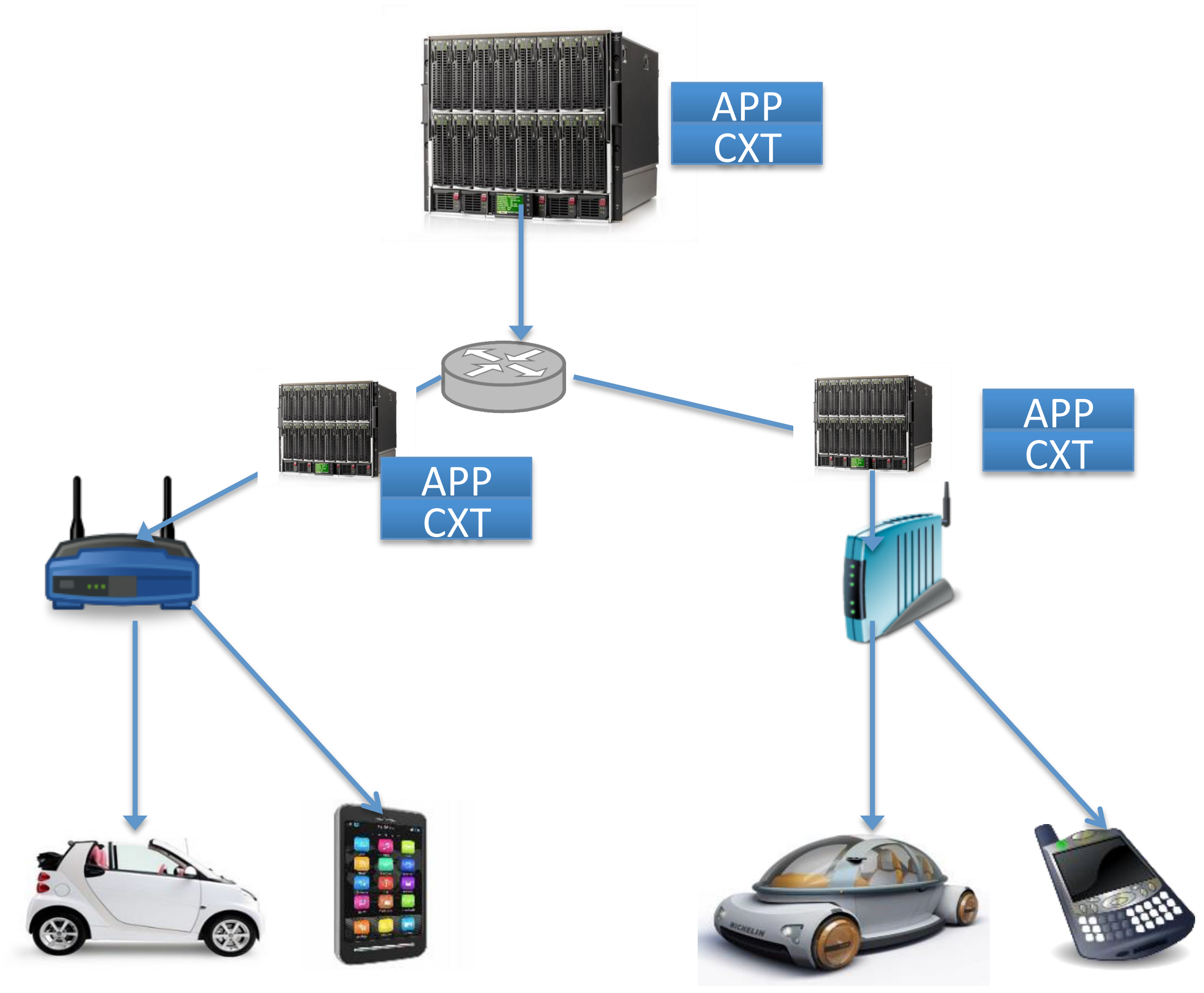
ACTIVATE_SERVICE()

CONTEXT

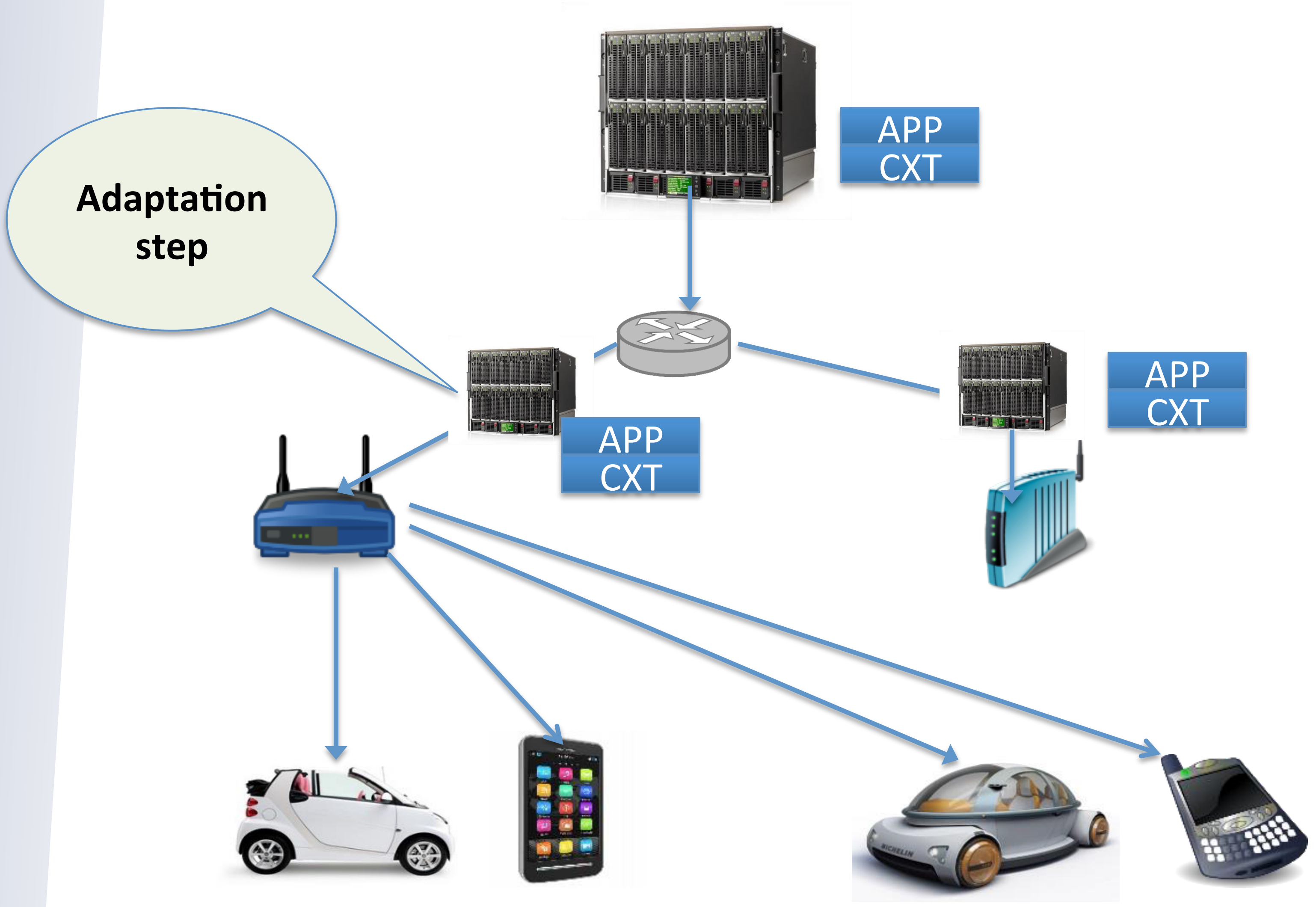
LOCAL SERVICES
HIERARCHY INFO
POLICIES



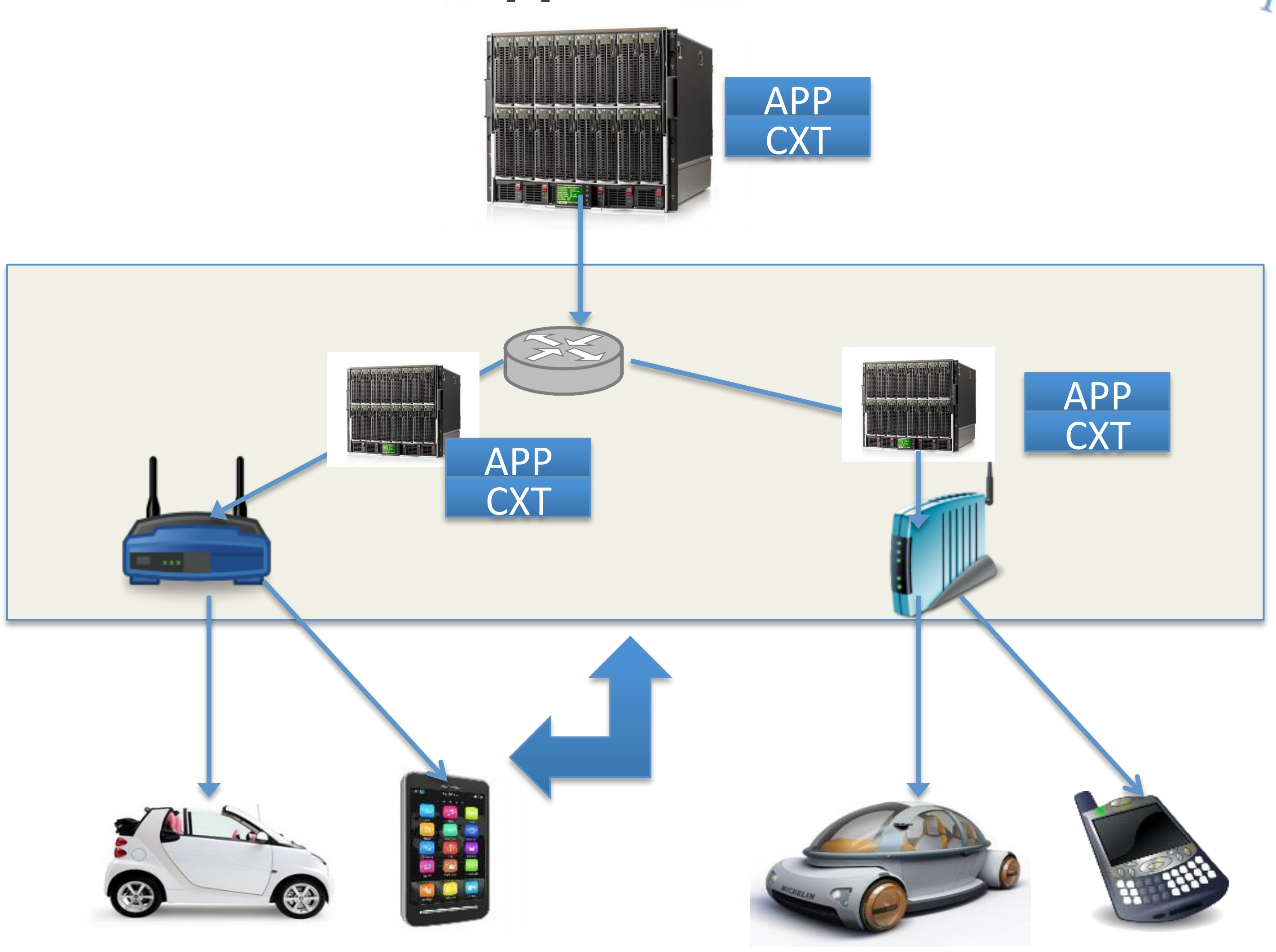
ARCHITECTURAL STYLE



Context-awareness: adaptation



Context-awareness: Coordinating Parallelism



WE ARE STILL LOST IN THE FOG …