CONTEXT-ORIENTED PROGRAMMING
ABSTRACTIONS FOR FOG

C. BODEI, P. DEGANO, G. FERRARI, L. GALLETTA
ML-CODA: A Context-Oriented Programming Language

FUNCTIONAL LANGUAGE

ML (KERNEL)

CONTEXT

DATALOG KNOWLEDGE BASE

ML-CODA

BODEI, CANCIANI, DEGANO, GALLETTA, SALVATORI, FERRARI
CONTEXT

ML (KERNEL)

ML-CODA

ABSTRACTION LAYER
VIRTUALIZATION OF THE
OPERATIONAL ENVIRONMENT

PHYSICAL ENV:
POSITION
CONNECTIVITY :
APP PREFS
PROFILES :

3
Context Dependent Binding

```plaintext
dlet txt= 
  getTxt ()
  when only_speech()
  in ...

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

CONTEXT
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

- Static Machinery (DFG@IEEE-TSE)
  - verify that dispatching mechanism always succeed
- Security Analysis (DBGS@JCS)
  - detect potential unsafe modifications
- 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
CONTEXT

TOPOLOGY OF STLs
LOCAL SERVICES
HIERARCHY INFO
CONTEXT

TOPOLOGY OF STLs
LOCAL SERVICES
HIERARCHY INFO

LOCAL & NON LOCAL

PUT_DATA()

COMPUTING & ORCHESTRATION
ORCHESTRATE(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 ...