Code Mobility

Konstantin Selyunin
e1228206@student.tuwien.ac.at
Miljenko Jakovljević
e1228206@student.tuwien.ac.at
Igor Pelesić
e0006828@student.tuwien.ac.at

December 2, 2012

Outline

Motivation

- Design code mobility system on ESE Board
- Practical experience
- Project management skills

Code mobility overview
Concept of code mobility

Concept of code mobility

Mobile agents

Meta-level knowledge

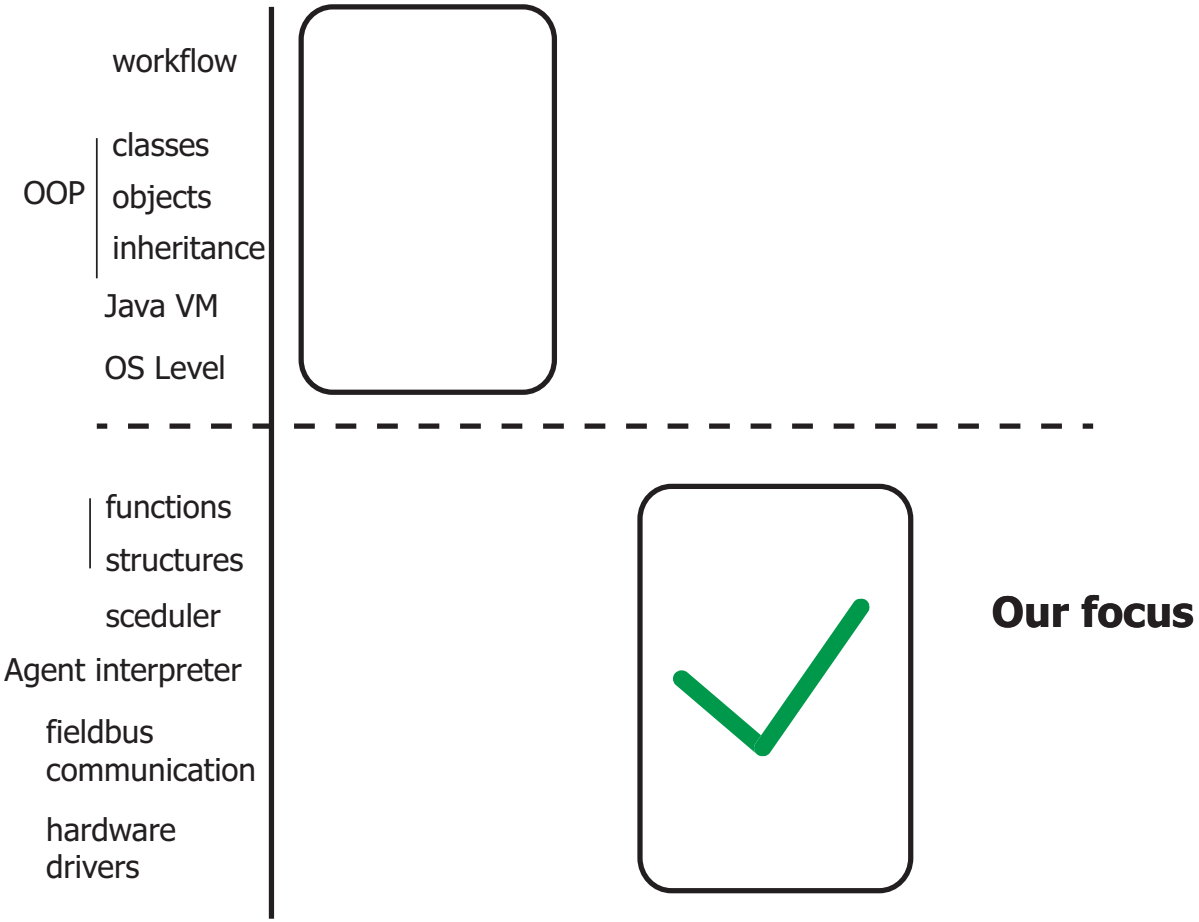
Strong and *weak* code mobility

Advantages of code mobility

Move code close to resources

Enable client customization of remote resources

Level of abstraction



Design challenges for the project

Processing gap
Performance
Memory management
Communication design

Requirements

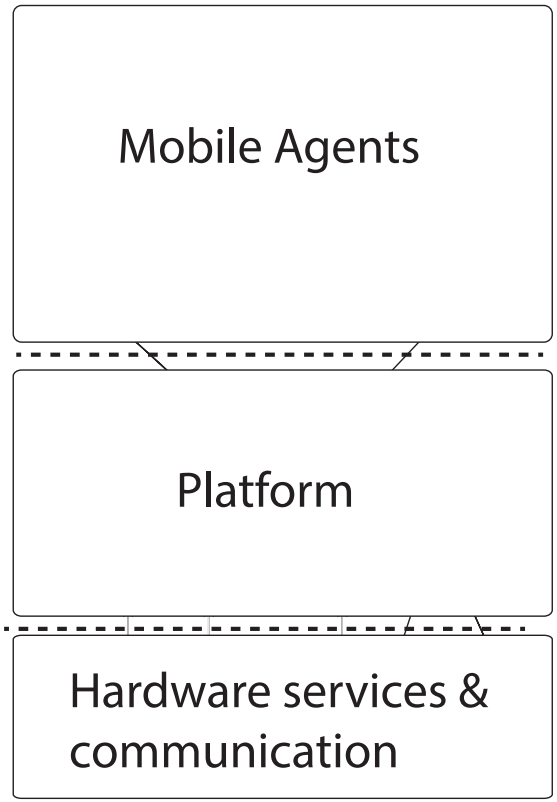
Design such a system that allows:

- Agents: up to 4 agents on one platform
- Platform: execute agents concurrently
- Communication: transfer agents *strong mobility* transfer messages between platforms

General overview

3 layered archtecture:

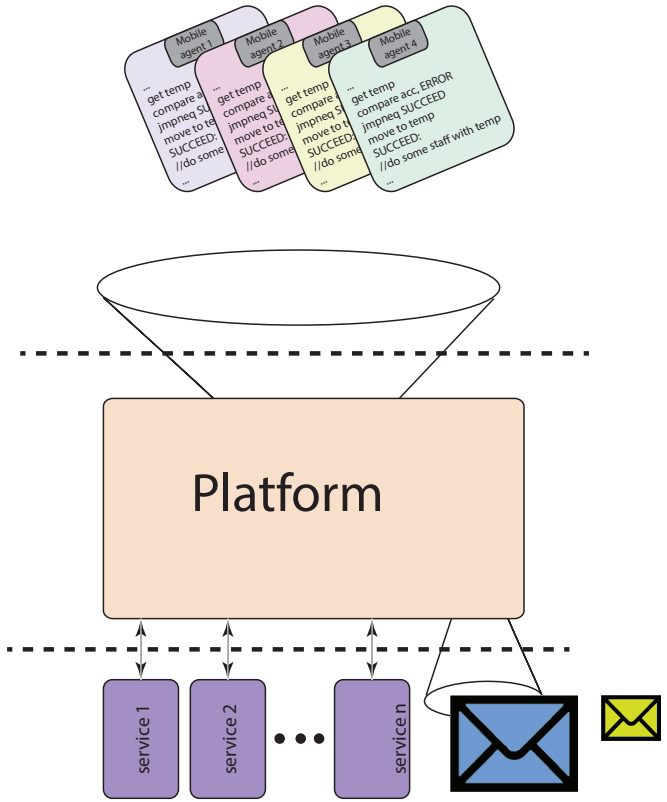
- Agent level
- Platform level
- communication & drivers



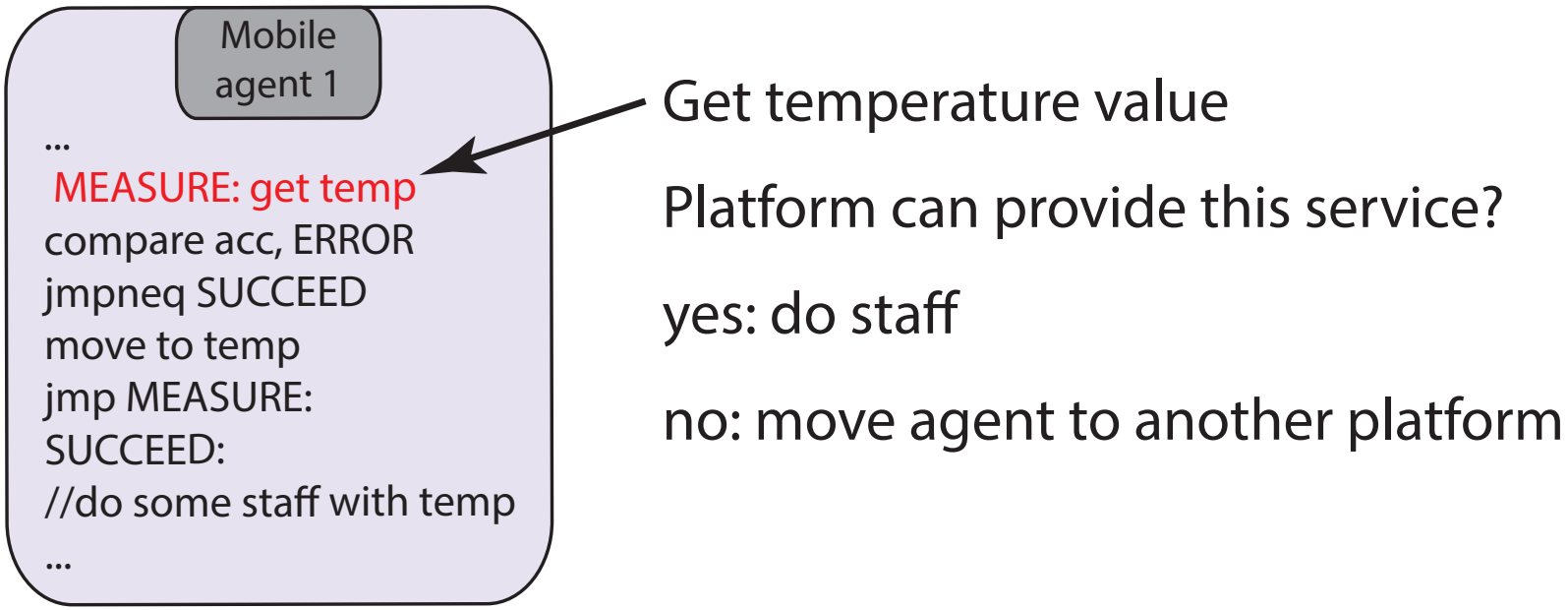
General overview

3 layered archtecture:

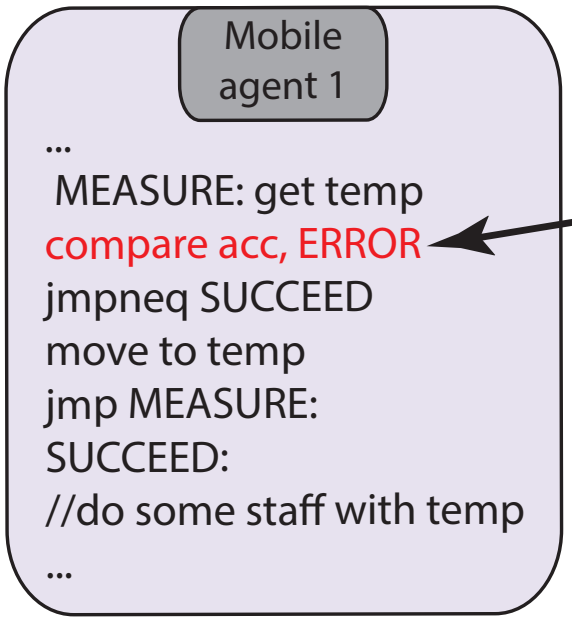
- Agent level
- Platform level
- communication & drivers



Agents


Mobile agent 1
...
MEASURE: get temp
compare acc, ERROR
jmpneq SUCCEED
move to temp
jmp MEASURE:
SUCCEED:
//do some staff with temp
...
Get temperature value
Platform can provide this service?
yes: do staff
no: move agent to another platform

Agents



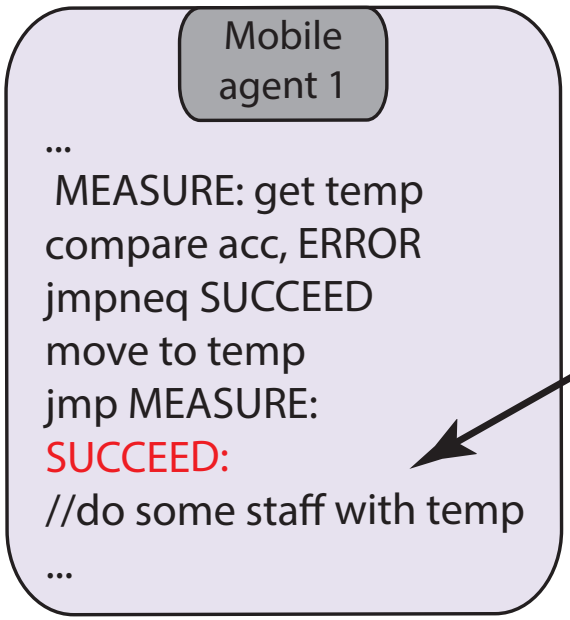
Get temperature value

Platform can provide this service?

yes: do staff

no: move agent to another platform

Agents



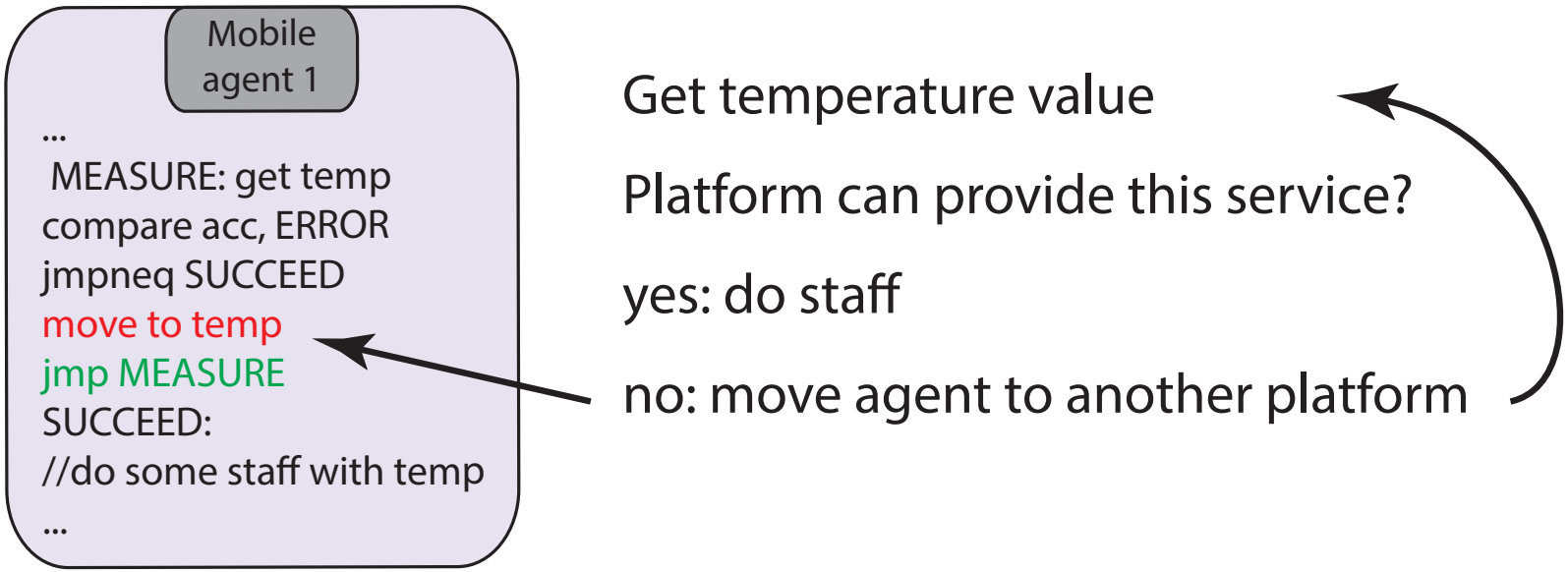
Get temperature value

Platform can provide this service?

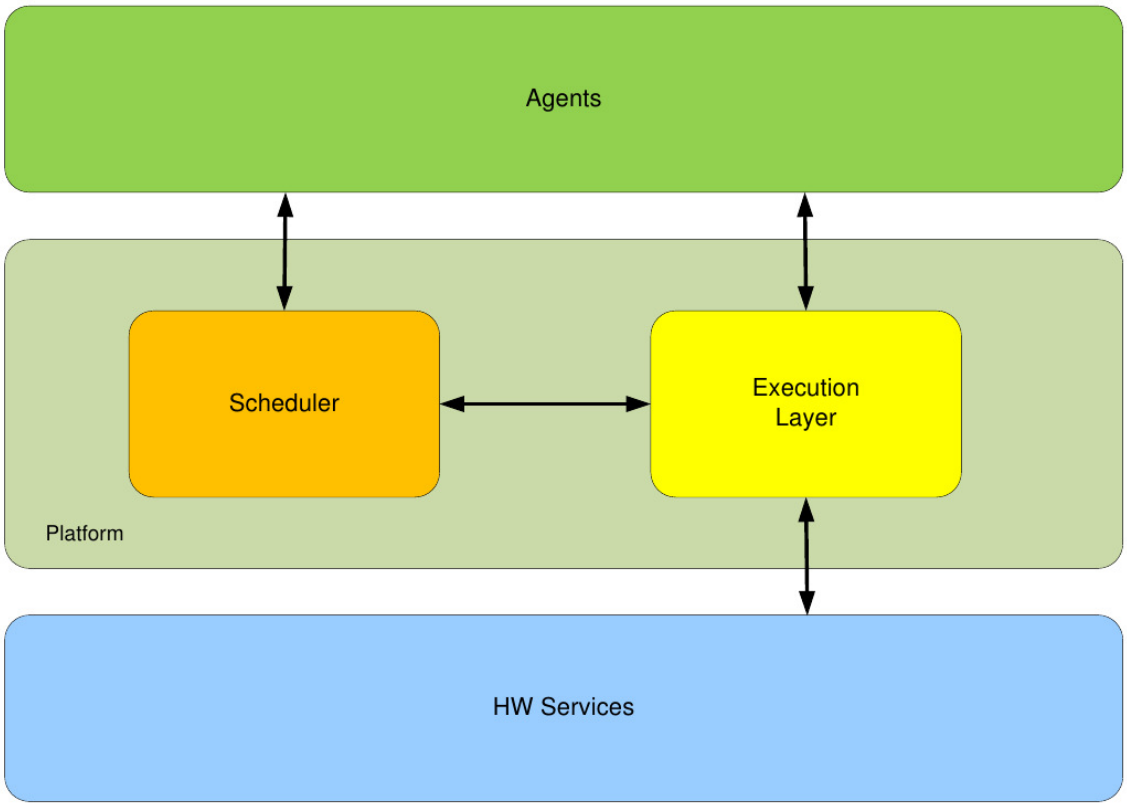
yes: do staff

no: move agent to another platform

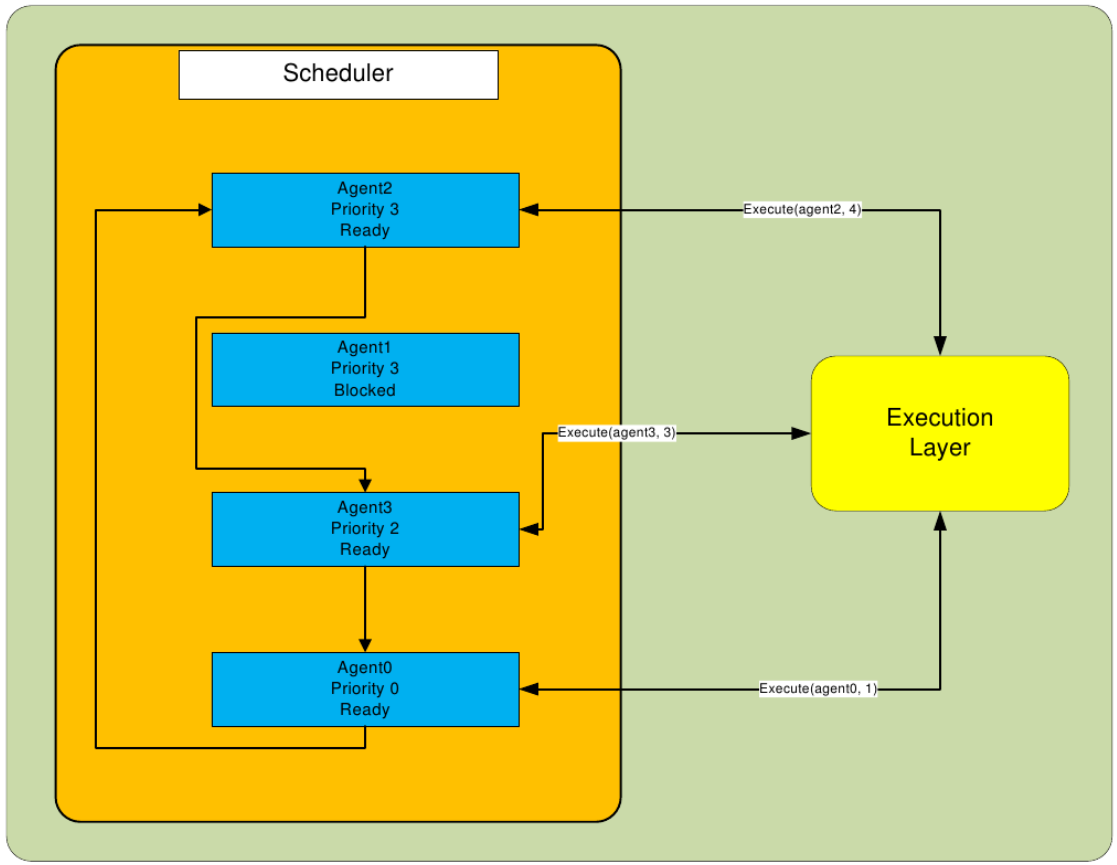
Agents


Mobile agent 1
...
MEASURE: get temp
compare acc, ERROR
jmpneq SUCCEED
move to temp
jmp MEASURE
SUCCEED:
//do some staff with temp
...
Get temperature value
Platform can provide this service?
yes: do staff
no: move agent to another platform

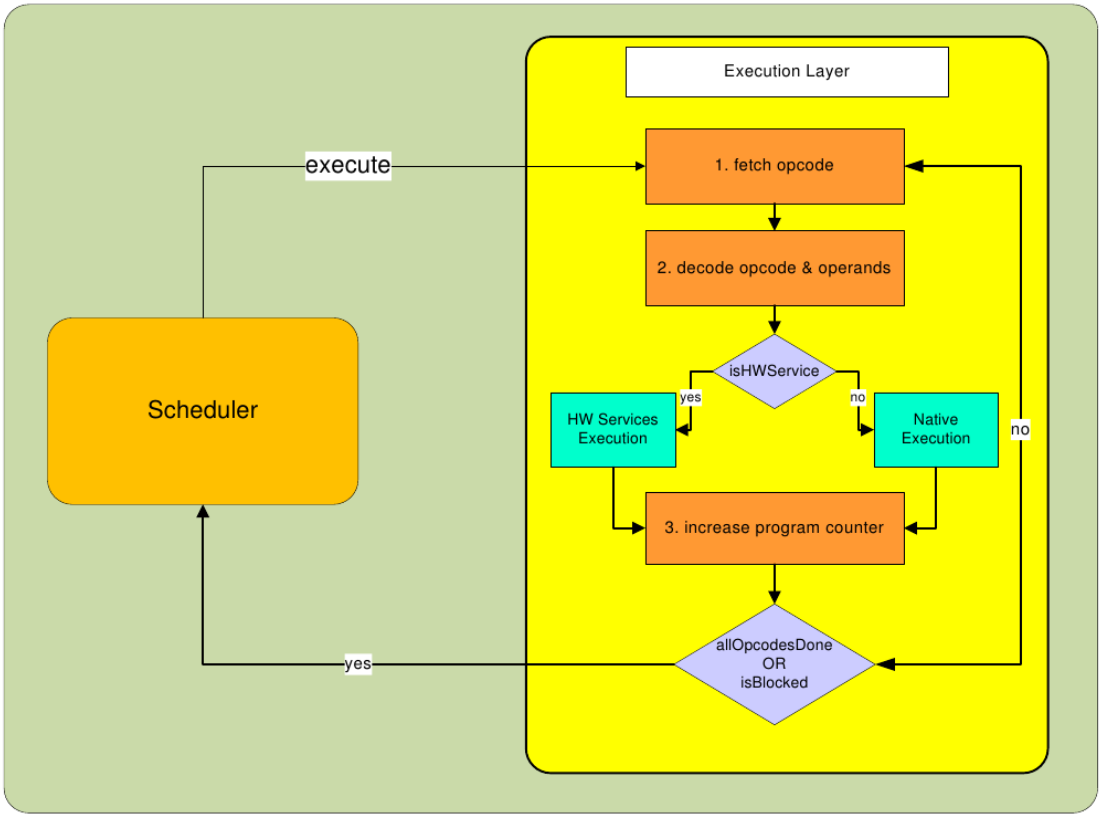
Platform



Scheduler


Scheduler
Agent2
Priority 3
Ready
Agent1
Priority 3
Blocked
Agent3
Priority 2
Ready
Agent0
Priority 0
Ready
Execute(agent2, 4)
Execute(agent3, 3)
Execute(agent0, 1)
Execution
Layer

Execution Layer



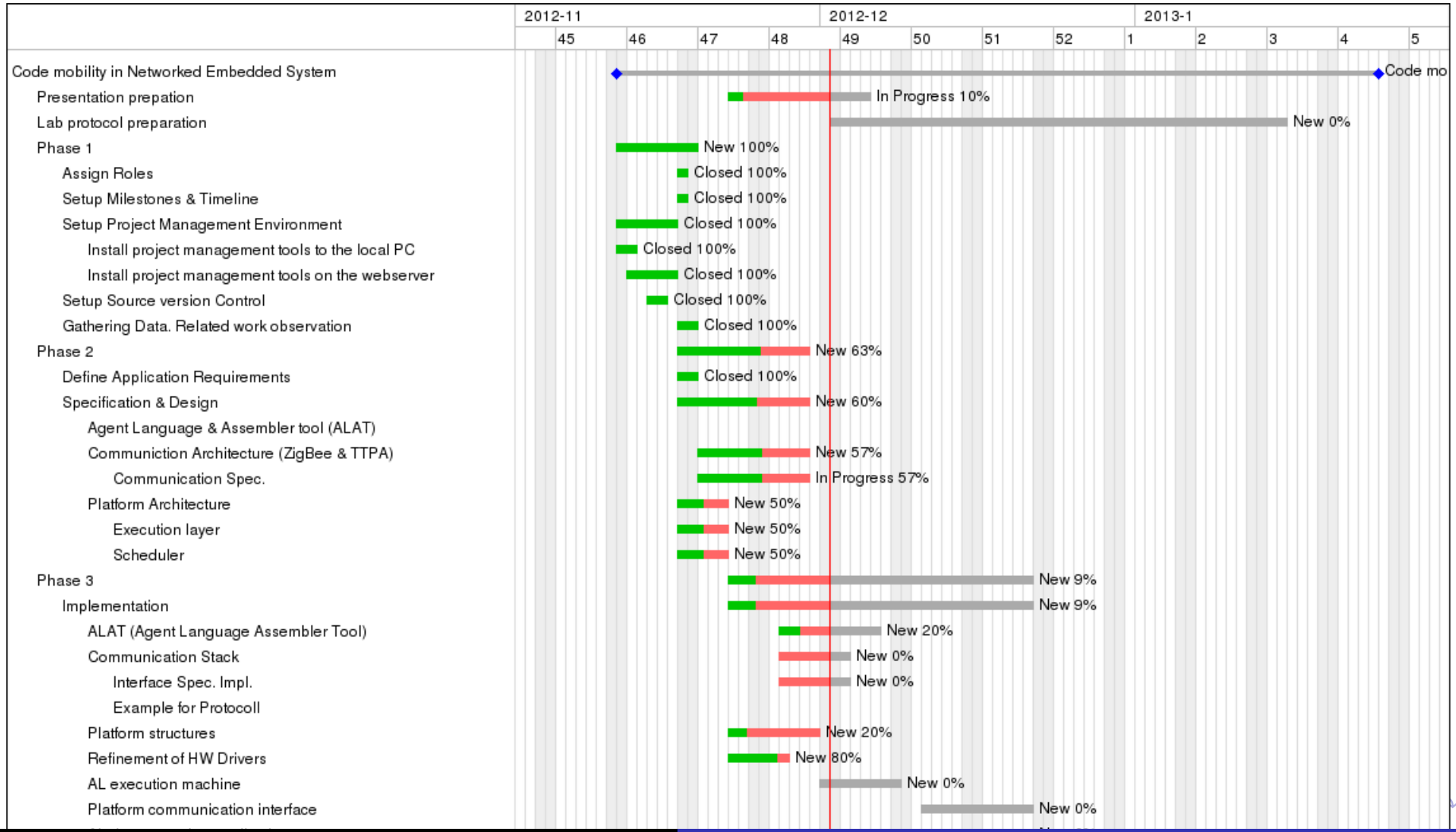
Communication

Milestones

- Phase 1. Product outline and information gathering
- Phase 2. Application requirements and specification
- Phase 3. Implementation
- Phase 4. Validation and analysis

Workpackages

Gantt diagram



Tools

Version control git github Project management redmine Code generation SCADE