Project's holder: Olivier Gruber

POLYTECH GRENOBLE

Implementing services upon Quark

AIR Project - Defense

Lucas CHALOYARD - Elias EL YANDOUZI

18/03/2022

Project objectives

From understanding to implementing

Discovering code base and concepts

Implement use cases upon Quark

Validate concepts and implementation of Quark

Group service

Used by TOM for dynamic group

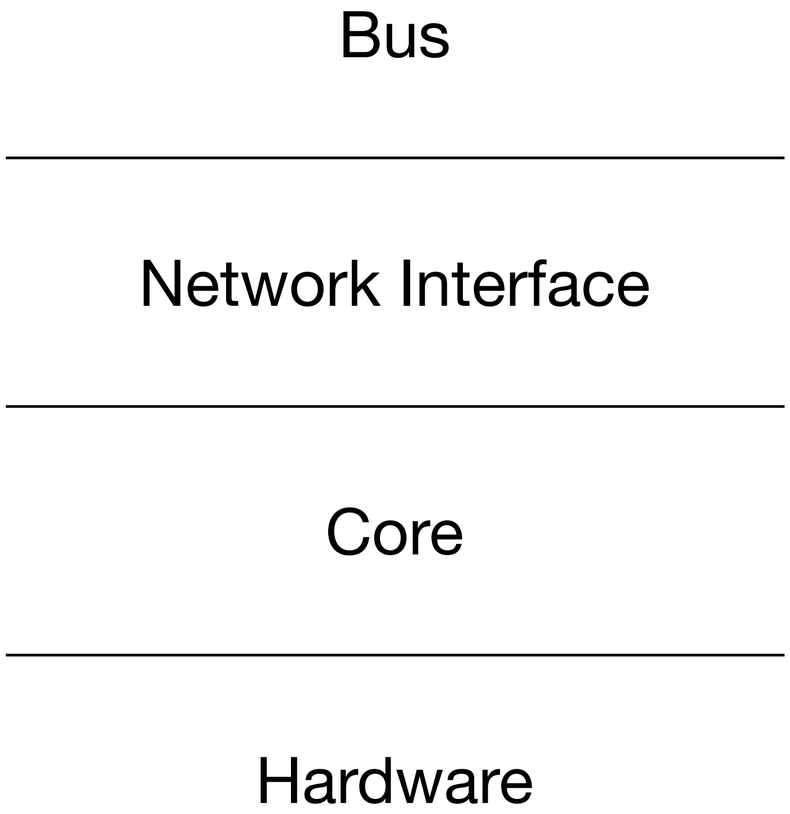
TOM

Well-known use case
Does stress the layers below
Self-testable

Quark, the bare-metal platform

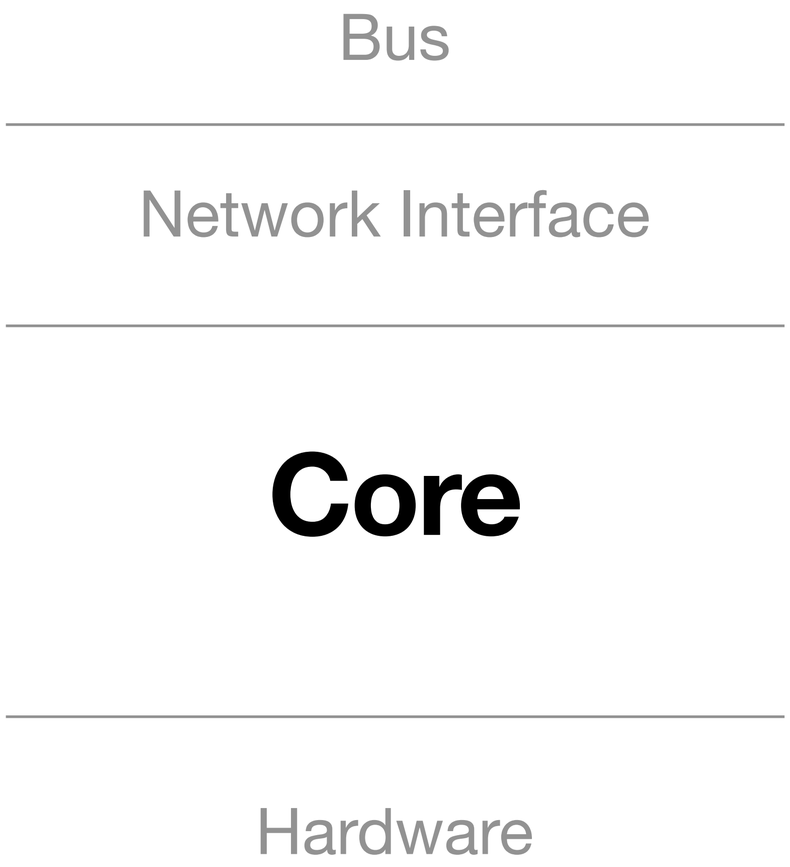
Software architecture

A matter of layers



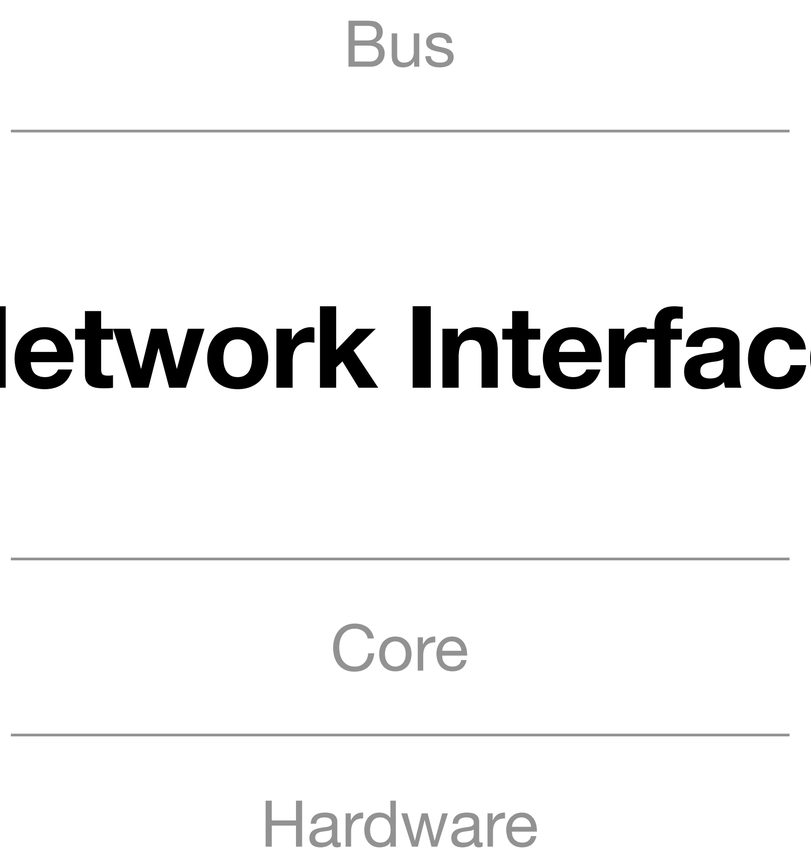
Software architecture

- Deals with memory management
- Event-oriented paradigm
  - An event-pump per core
  - Run-to-completion



Software architecture

- Provides access to network
- Enables frames transmission
- Brings the concept of nodes and network addresses



Software architecture

Two key concepts

Channel / Record

An event-driven socket

Record flow

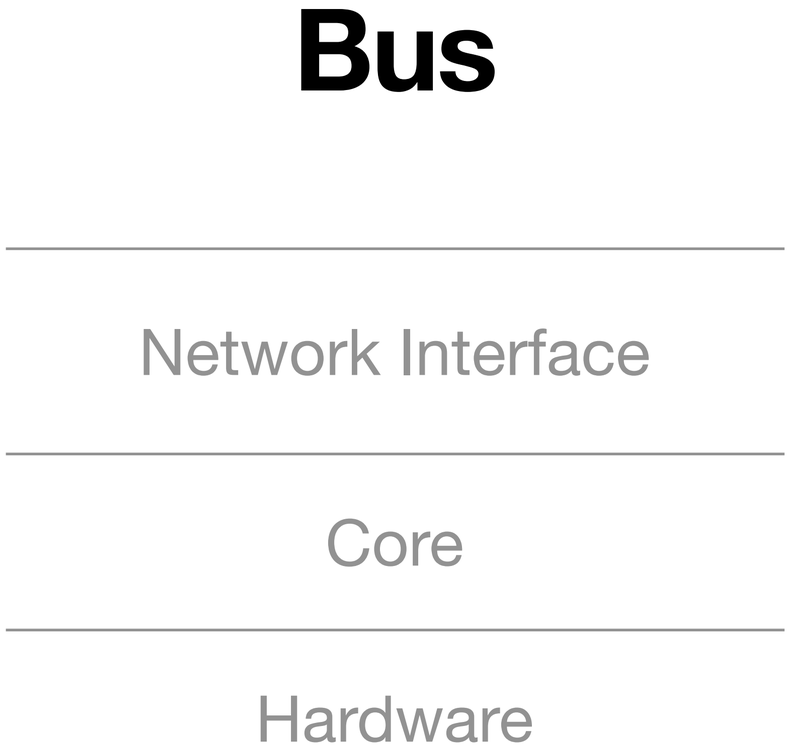
FIFO / Lossless

Protocol / Query

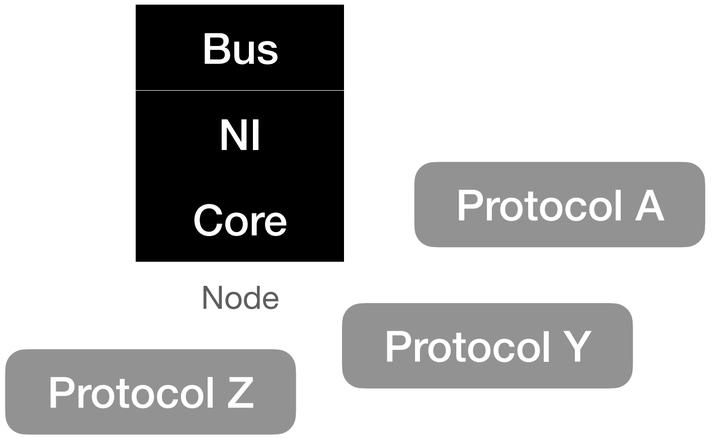
Defined by ID and version (instead of port)

Recorded on a node

Matching protocols give channels



Software architecture



Two key concepts

Channel / Record

An event-driven socket

Record flow

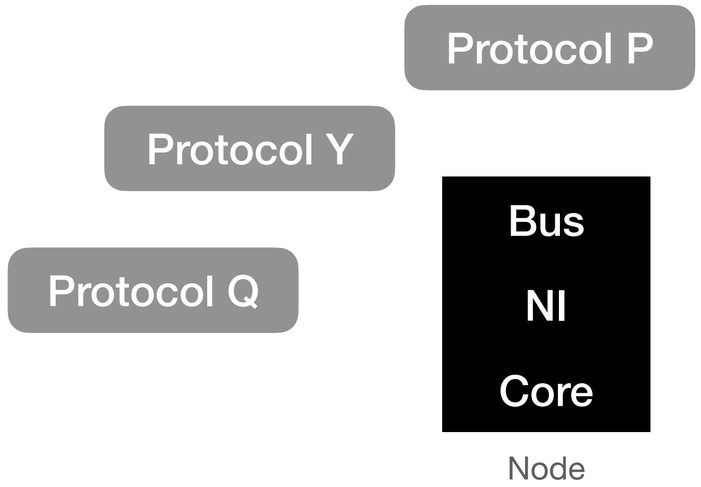
FIFO / Lossless

Protocol / Query

Defined by ID and version (instead of port)

Recorded on a node

Matching protocols give channels



Software architecture

Two key concepts

Channel / Record

An event-driven socket

Record flow

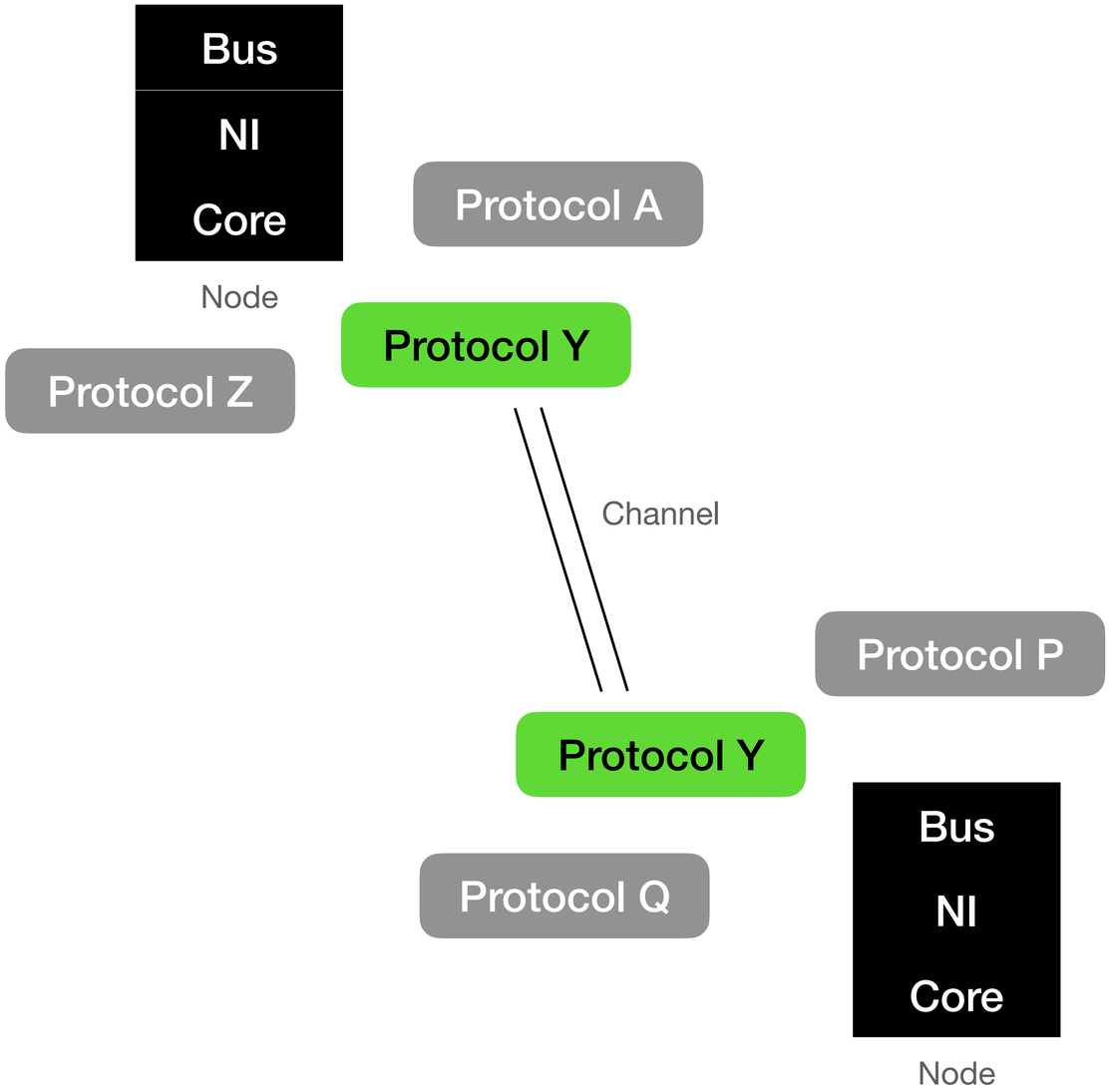
FIFO / Lossless

Protocol / Query

Defined by ID and version (instead of port)

Recorded on a node

Matching protocols give channels



Execution Architecture

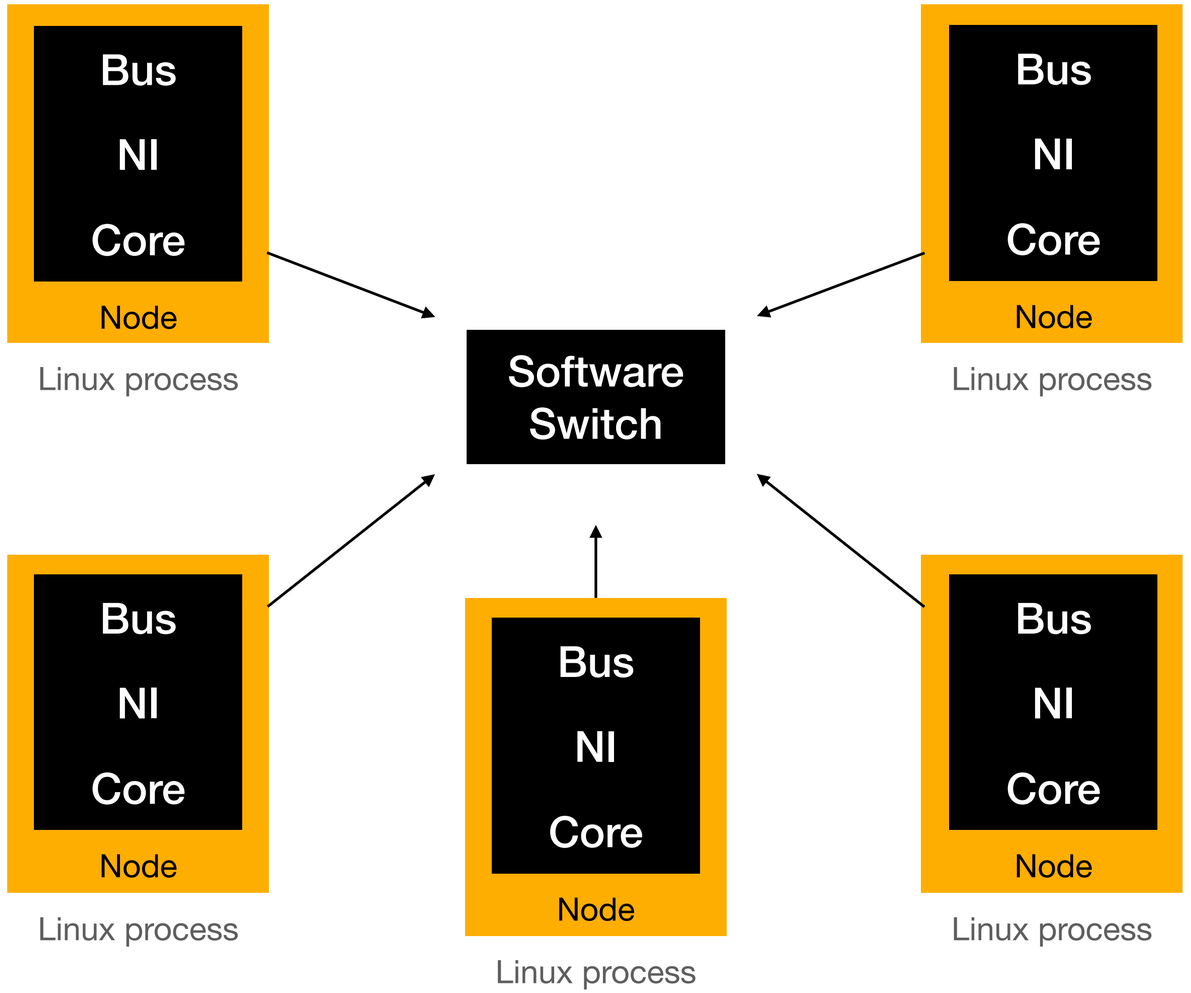
A multi-target project

- Quark is emulated in a Linux process
- Every processes can be interconnected

Execution Architecture

A multi-target project

- Quark is emulated in Linux process
- Every processes are interconnected



Distributed system example

A look on the carried work

Group service

What is it?

- Groups are meeting place for peers
- A special peer by group - coordinator
  - Get notified of joining peer
  - Notify the group service of leaving peer
- Fault tolerant

Group service

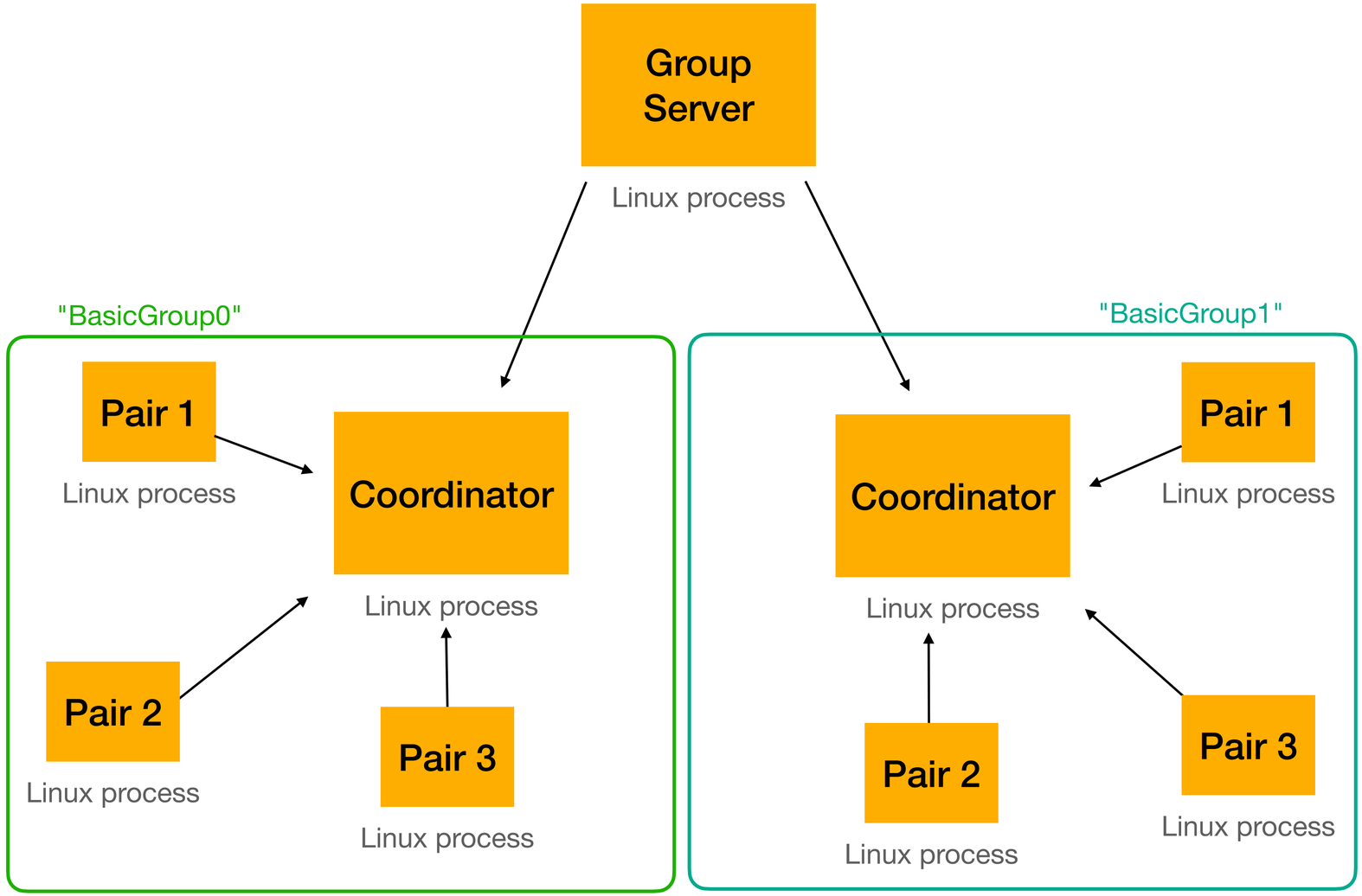
Development phase

- Design phase
  - Difficulty to separate the layers
  - Asynchronous environment
- Implementation phase
  - Coherency between API and environment
- Conclusion
  - Features implemented and tested
  - Well fitting API

Group service

Demonstration 1

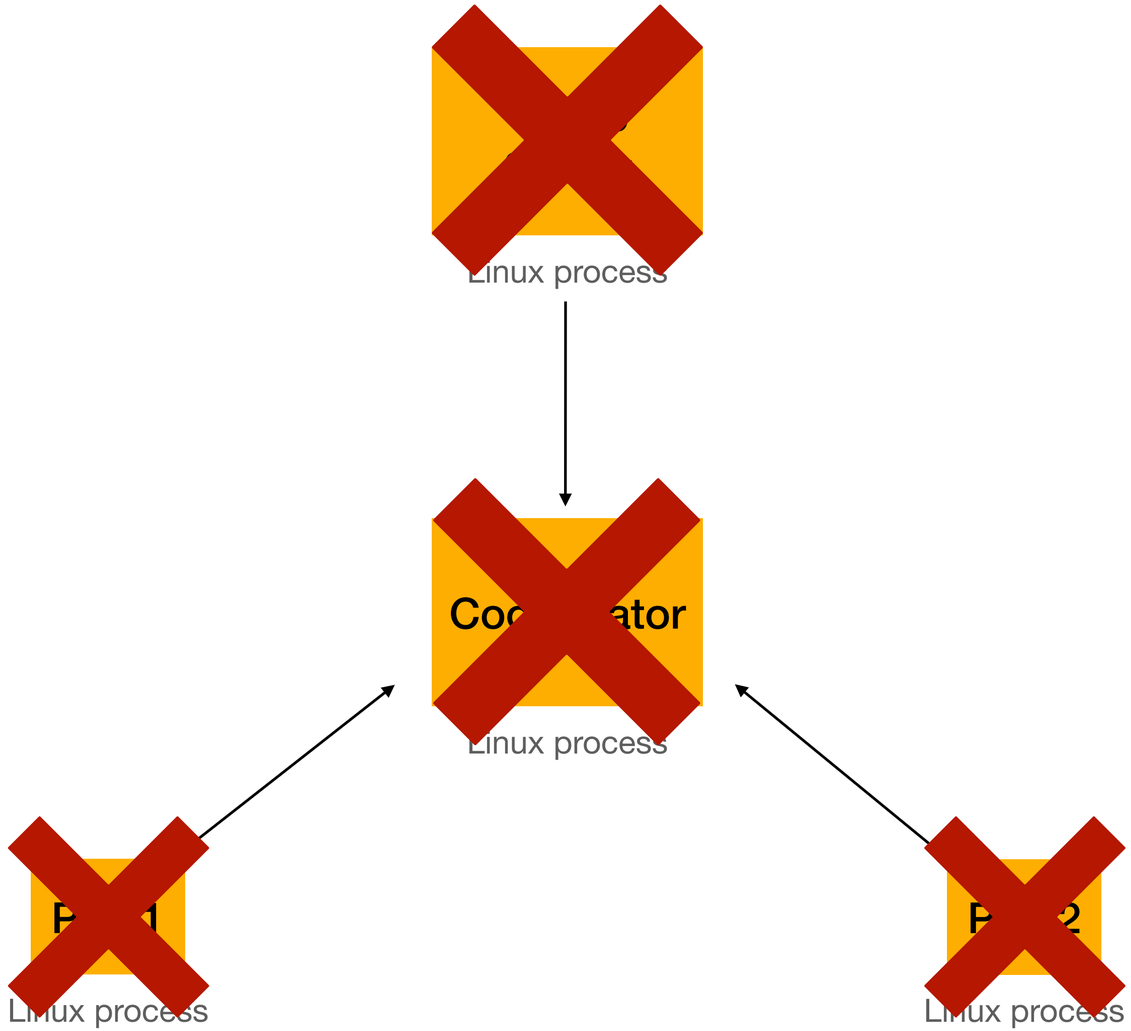
Group creation and join



Group service

Demonstration 2

Fault tolerance



Totally Ordered Multicast

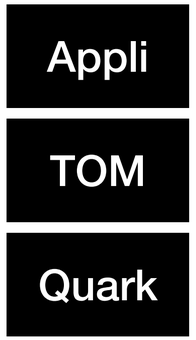
What is it?

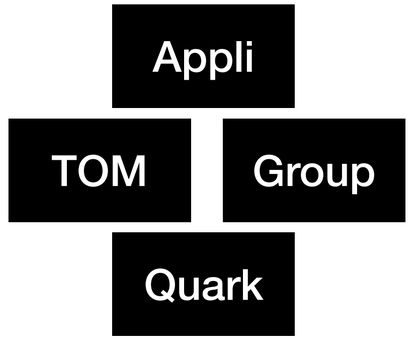
- A well-known use case which stress layers below
- Designed to be fault tolerant
- Self-testable

Totally Ordered Multicast

What is it?

- A well-known use case which stress layers below
- Designed to be fault tolerant
- Self-testable



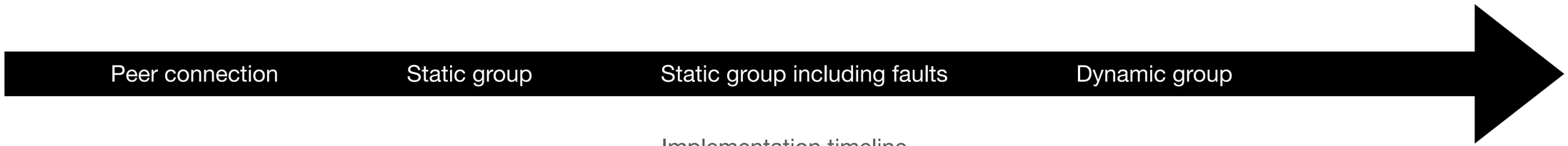


Can rely on Group service

Totally Ordered Multicast

Development phase

- Design and discussion with tech lead to define TOM semantic, including faults
- Incremental implementation



Implementation timeline

Totally Ordered Multicast

Demonstration

Static group - no fault

Static group - fault included

Totally Ordered Multicast

Demonstration

Static group - no fault

1. Connection phase
2. Message delivered according total order

```
peer[0:0x2000036e0]: found the TOM service. (prot: 0x200003904)
peer[0:0x2000036e0]: found the TOM service. (prot: 0x200004554)
peer[0:0x2000036e0]: found the TOM service. (prot: 0x200004e4c)
peer[1:0x200004330]: found the TOM service. (prot: 0x200004554)
peer[1:0x200004330]: found the TOM service. (prot: 0x200004e4c)
peer[2:0x200004c28]: found the TOM service. (prot: 0x200004e4c)
Peer[0] joined the group and has established all the required connection.
Peer[1] joined the group and has established all the required connection.
Peer[2] joined the group and has established all the required connection.
Message delivered from peer [2] - [ID:0 | TS:0] : Hello
Message delivered from peer [2] - [ID:1 | TS:0] : Hola
Message delivered from peer [2] - [ID:2 | TS:0] : Ciao
Message delivered from peer [1] - [ID:0 | TS:0] : Hello
Message delivered from peer [1] - [ID:1 | TS:0] : Hola
Message delivered from peer [1] - [ID:2 | TS:0] : Ciao
Message delivered from peer [0] - [ID:0 | TS:0] : Hello
Message delivered from peer [0] - [ID:1 | TS:0] : Hola
Message delivered from peer [0] - [ID:2 | TS:0] : Ciao
Message delivered from peer [0] - [ID:0 | TS:1] : Hello
Message delivered from peer [2] - [ID:0 | TS:1] : Hello
Message delivered from peer [1] - [ID:0 | TS:1] : Hello
Message delivered from peer [0] - [ID:1 | TS:1] : Hola
Message delivered from peer [2] - [ID:1 | TS:1] : Hola
Message delivered from peer [1] - [ID:1 | TS:1] : Hola
Message delivered from peer [0] - [ID:2 | TS:1] : Ciao
Message delivered from peer [0] - [ID:0 | TS:2] : Hello
Message delivered from peer [0] - [ID:1 | TS:2] : Hola
Message delivered from peer [2] - [ID:2 | TS:1] : Ciao
Message delivered from peer [2] - [ID:0 | TS:2] : Hello
Message delivered from peer [2] - [ID:1 | TS:2] : Hola
Message delivered from peer [2] - [ID:2 | TS:2] : Ciao
Message delivered from peer [1] - [ID:2 | TS:1] : Ciao
Message delivered from peer [1] - [ID:0 | TS:2] : Hello
Message delivered from peer [1] - [ID:1 | TS:2] : Hola
Message delivered from peer [0] - [ID:2 | TS:2] : Ciao
Message delivered from peer [1] - [ID:2 | TS:2] : Ciao
Message delivered from peer [0] - [ID:0 | TS:3] : Hello
Message delivered from peer [2] - [ID:0 | TS:3] : Hello
Message delivered from peer [1] - [ID:0 | TS:3] : Hello
Message delivered from peer [2] - [ID:1 | TS:3] : Hola
Message delivered from peer [2] - [ID:2 | TS:3] : Ciao
Message delivered from peer [1] - [ID:1 | TS:3] : Hola
Message delivered from peer [1] - [ID:2 | TS:3] : Ciao
Message delivered from peer [0] - [ID:1 | TS:3] : Hola
Message delivered from peer [0] - [ID:2 | TS:3] : Ciao
Message delivered from peer [0] - [ID:0 | TS:4] : Hello
Message delivered from peer [2] - [ID:0 | TS:4] : Hello
Message delivered from peer [1] - [ID:0 | TS:4] : Hello
Message delivered from peer [1] - [ID:1 | TS:4] : Hola
Message delivered from peer [1] - [ID:2 | TS:4] : Ciao
Message delivered from peer [0] - [ID:1 | TS:4] : Hola
Message delivered from peer [0] - [ID:2 | TS:4] : Ciao
Message delivered from peer [2] - [ID:1 | TS:4] : Hola
Message delivered from peer [2] - [ID:2 | TS:4] : Ciao
```

Totally Ordered Multicast

Demonstration

Static group - no fault

Static group - fault included

Totally Ordered Multicast

Demonstration

Static group - fault included

1. *GoodBye0* indicates *peer[0]* death
2. *GoodBye0* message reemitted

```
peer[0:0x2000036e0]: found the TOM service. (prot: 0x200004554)
peer[0:0x2000036e0]: found the TOM service. (prot: 0x200004e4c)
peer[1:0x200004330]: found the TOM service. (prot: 0x200004554)
peer[1:0x200004330]: found the TOM service. (prot: 0x200004e4c)
peer[2:0x200004c28]: found the TOM service. (prot: 0x200004e4c)
Peer[0] joined the group and has established all the required connection.
Peer[1] joined the group and has established all the required connection.
Peer[2] joined the group and has established all the required connection.
Message delivered from peer [2] - [ID:0 | TS:0] : Hello
Message delivered from peer [1] - [ID:0 | TS:0] : Hello
Message delivered from peer [0] - [ID:0 | TS:0] : Hello
Message delivered from peer [2] - [ID:0 | TS:1] : Hello
Message delivered from peer [1] - [ID:0 | TS:1] : Hello
Message delivered from peer [0] - [ID:0 | TS:1] : Hello
Message delivered from peer [2] - [ID:0 | TS:2] : Hello
Message delivered from peer [1] - [ID:0 | TS:2] : Hello
Message delivered from peer [0] - [ID:0 | TS:2] : Hello
Message delivered from peer [2] - [ID:0 | TS:3] : Hello
Message delivered from peer [1] - [ID:0 | TS:3] : Hello
Message delivered from peer [0] - [ID:0 | TS:3] : Hello
Message delivered from peer [1] - [ID:0 | TS:4] : GoodBye0
Message delivered from peer [2] - [ID:0 | TS:4] : GoodBye0
Message delivered from peer [2] - [ID:2 | TS:5] : Hello2
Message delivered from peer [1] - [ID:2 | TS:5] : Hello2
Message delivered from peer [2] - [ID:2 | TS:6] : Hello2
Message delivered from peer [1] - [ID:2 | TS:6] : Hello2
Message delivered from peer [2] - [ID:1 | TS:7] : GoodBye1
Message delivered from peer [2] - [ID:2 | TS:7] : Hello2
Message delivered from peer [2] - [ID:2 | TS:8] : Hello2
Message delivered from peer [2] - [ID:2 | TS:9] : Hello2
```

Totally Ordered Multicast

Demonstration

Static group - fault included

1. *GoodBye0* indicates *peer[0]* death
2. *GoodBye0* message reemitted

```
static
void test_death(struct peer* peer){
  if(peer->id == 0 && peer->clock < 4)
    tom_send(peer, (uint8_t*)"Hello\n\0", 7);
  else if (peer->id == 0 && peer->clock == 4){
    die(peer, 1);
    p0_dead = true;
  } else if (peer->id == 1 && peer->clock == 7){
    die(peer, 1);
  }

  if (p0_dead && peer->id == 2 && peer->clock < 10)
    tom_send(peer, (uint8_t*)"Hello2\n\0", 8);
}
```

Totally Ordered Multicast

Demonstration

Static group - fault included

1. *GoodBye0* indicates *peer[0]* death
2. *GoodBye0* message reemitted
3. Death doesn’t stop delivery

```
peer[0:0x2000036e0]: found the TOM service. (prot: 0x200004554)
peer[0:0x2000036e0]: found the TOM service. (prot: 0x200004e4c)
peer[1:0x200004330]: found the TOM service. (prot: 0x200004554)
peer[1:0x200004330]: found the TOM service. (prot: 0x200004e4c)
peer[2:0x200004c28]: found the TOM service. (prot: 0x200004e4c)
Peer[0] joined the group and has established all the required connection.
Peer[1] joined the group and has established all the required connection.
Peer[2] joined the group and has established all the required connection.
Message delivered from peer [2] - [ID:0 | TS:0] : Hello
Message delivered from peer [1] - [ID:0 | TS:0] : Hello
Message delivered from peer [0] - [ID:0 | TS:0] : Hello
Message delivered from peer [2] - [ID:0 | TS:1] : Hello
Message delivered from peer [1] - [ID:0 | TS:1] : Hello
Message delivered from peer [0] - [ID:0 | TS:1] : Hello
Message delivered from peer [2] - [ID:0 | TS:2] : Hello
Message delivered from peer [1] - [ID:0 | TS:2] : Hello
Message delivered from peer [0] - [ID:0 | TS:2] : Hello
Message delivered from peer [2] - [ID:0 | TS:3] : Hello
Message delivered from peer [1] - [ID:0 | TS:3] : Hello
Message delivered from peer [0] - [ID:0 | TS:3] : Hello
Message delivered from peer [1] - [ID:0 | TS:4] : GoodBye0
Message delivered from peer [2] - [ID:0 | TS:4] : GoodBye0
Message delivered from peer [2] - [ID:2 | TS:5] : Hello2
Message delivered from peer [1] - [ID:2 | TS:5] : Hello2
Message delivered from peer [2] - [ID:2 | TS:6] : Hello2
Message delivered from peer [1] - [ID:2 | TS:6] : Hello2
Message delivered from peer [2] - [ID:1 | TS:7] : GoodBye1
Message delivered from peer [2] - [ID:2 | TS:7] : Hello2
Message delivered from peer [2] - [ID:2 | TS:8] : Hello2
Message delivered from peer [2] - [ID:2 | TS:9] : Hello2
```

Project’s environment

Technologies and language

80’s style environment

- C bare-metal, no library
- GDB for debugging, Valgrind for memory
- Makefile inspired by Linux kernel

Project management

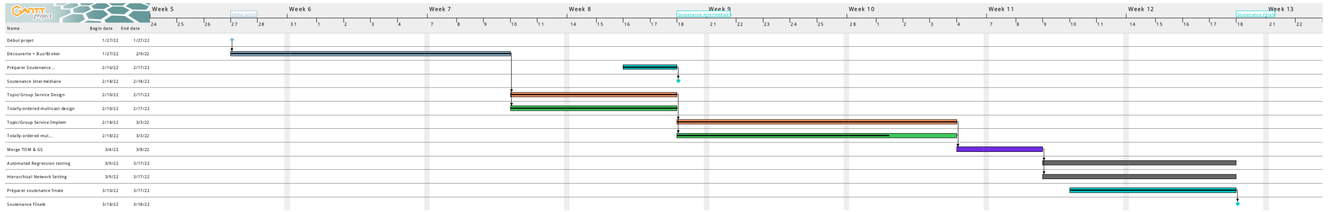
From organisation to stats

Quark Project

Mar 17, 2022

Gantt Chart

4



Name	Begin date	End date
Début projet	1/27/22	1/27/22
Découverte + Bus/Broker	1/27/22	2/9/22
Préparer Soutenance...	2/16/22	2/17/22
Soutenance Intermédiaire	2/18/22	2/18/22
Topic/Group Service Design	2/10/22	2/17/22
Totally-ordered multicast design	2/10/22	2/17/22
Topic/Group Service Implem	2/18/22	3/3/22
Totally-ordered mul...	2/18/22	3/3/22
Merge TOM & GS	3/4/22	3/8/22
Automated Regression testing	3/9/22	3/17/22
Hierarchical Network Setting	3/9/22	3/17/22
Préparer soutenance finale	3/10/22	3/17/22
Soutenance FInale	3/18/22	3/18/22

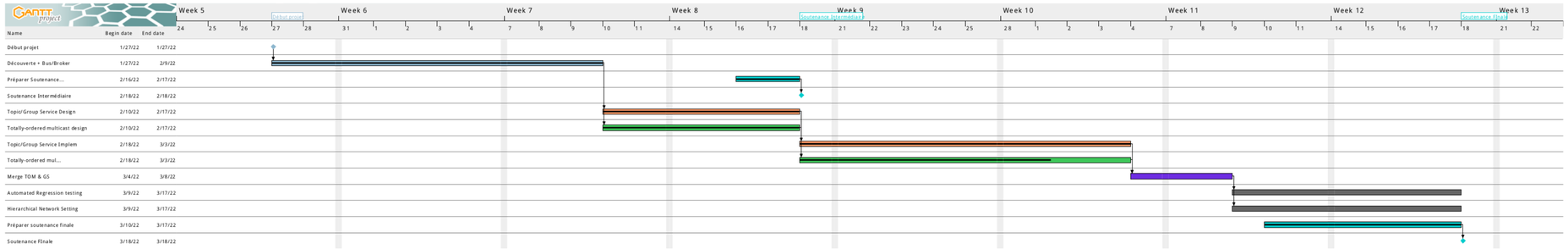
Quark Project

Mar 17, 2022

4

Gantt Chart

Name	Begin date	End date
Début projet	1/27/22	1/27/22
Découverte + Bus/Broker	1/27/22	2/9/22
Préparer Soutenance...	2/16/22	2/17/22
Soutenance Intermédiaire	2/18/22	2/18/22
Topic/Group Service Design	2/10/22	2/17/22
Totally-ordered multicast design	2/10/22	2/17/22
Topic/Group Service Implem	2/18/22	3/3/22
Totally-ordered mul...	2/18/22	3/3/22
Merge TOM & GS	3/4/22	3/8/22
Automated Regression testing	3/9/22	3/17/22
Hierarchical Network Setting	3/9/22	3/17/22
Préparer soutenance finale	3/10/22	3/17/22
Soutenance FInale	3/18/22	3/18/22



Project management

Internal organisation

- Bi-weekly meetings
- Use of git
  - Personal branches
  - Pull request to merge fix

Conclusion

Lessons learned

- Quark’s concepts validated and implementation strengthened
- Understood how much a design is valuable
- Dealt with an irregular paradigm and learnt a new way of thinking