# IMW: Middleware

Stephane Genaud

December 18, 2015

# Outline

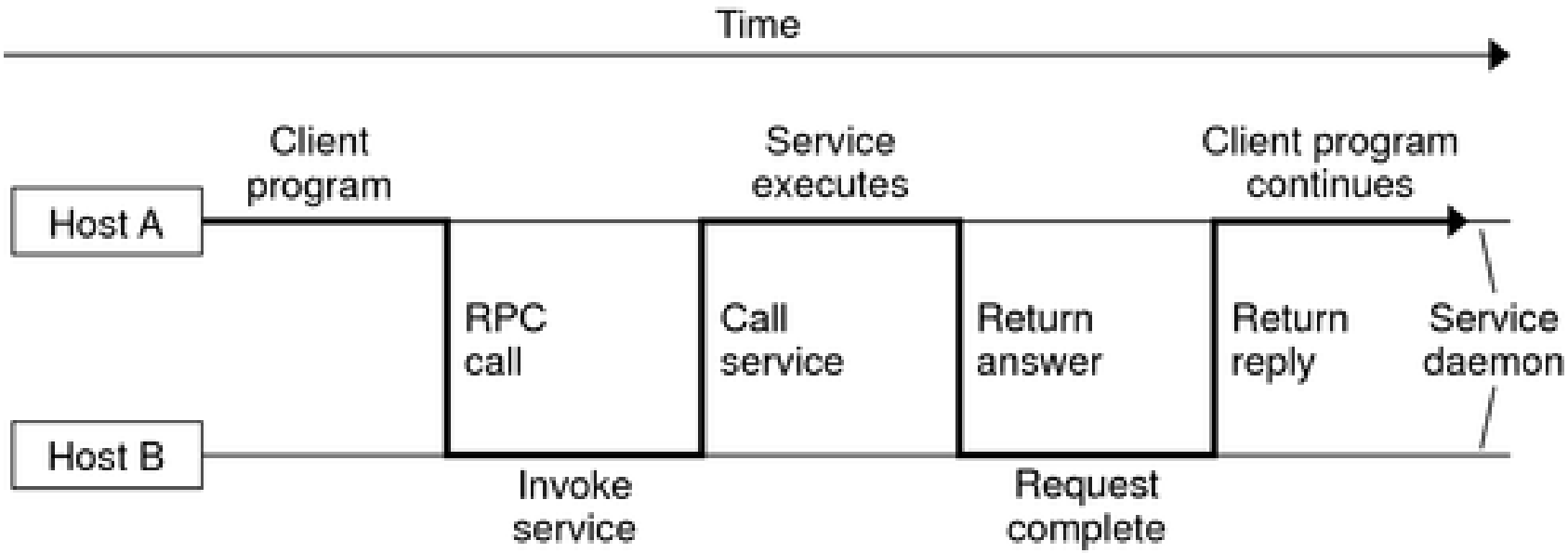
# Table of Contents

# Technologies for Distributed Systems

- Extremely fast evolution since 1985: about a technology every 5 years.
- Implementations adapt to up-to-date technology
  e.g If networks go faster, it is possible to convey bigger messages.
  If the cost of some hardware becomes low, no need to spare it.

# Technologies change but ... concepts stay

- Client-server is the central concept:
  The client can make a request at any time,
  The server permanently waits for incoming requests



# Middleware: definition

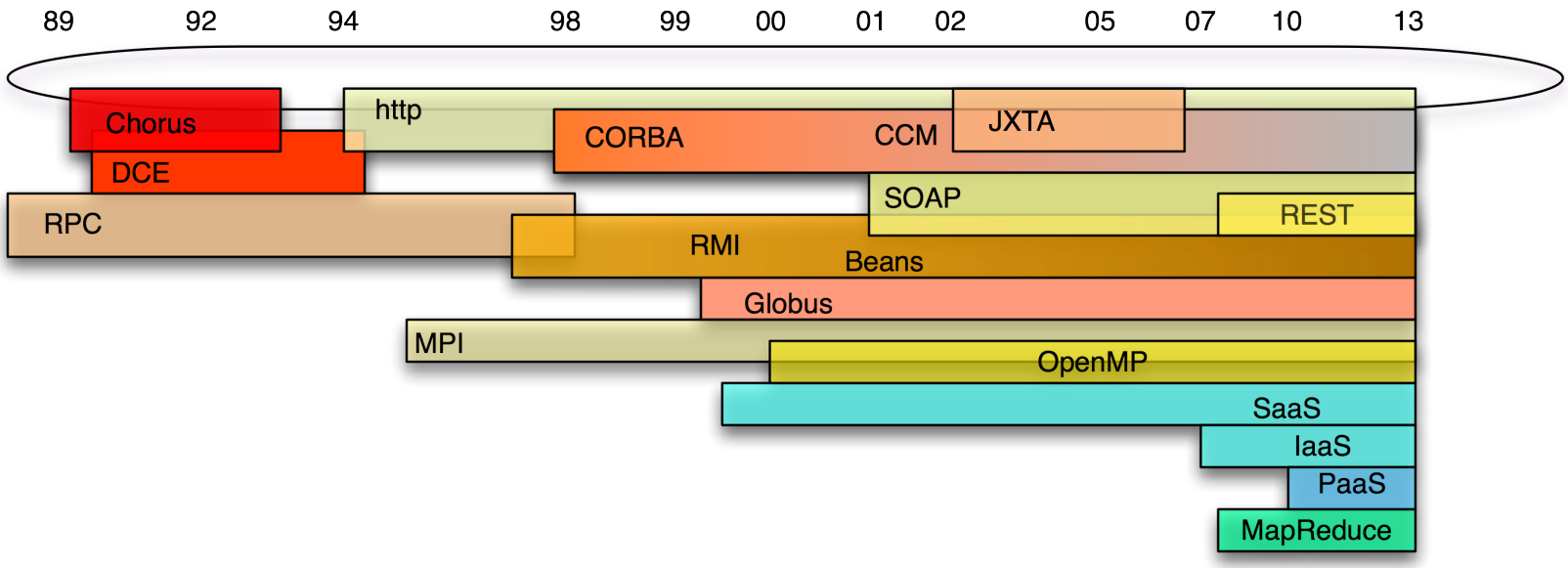
## What is middleware?

- A sofware layer between the OS and the application allowing a set of distributed computers to communicate in a standardized way.

## Middleware provides inter-machines communication facilities,

but may also include services, such as authentification services, resource directories, distributed file catalogs, ...

# A Time-line of technologies



# Principle Design Choices

- Abstraction vs. Performance
- Interoperability
- Versatility

# Abstraction

## Abstraction of communication primitives

- too low level: rapidly obsolete, lower programming productivity
- too high level: difficult to optimize for performance

## Abstraction Trade-off

- independent from the architecture: execute across different systems without source code modification
- Hide details related to communication/synchronization management (e.g `Remote Procedure Calls` more abstract than `sockets`)

# Interoperability

## Machine-independent

e.g Sun RPC

## OS-independent

e.g Java-RMI

## Language-independent

e.g Corba, SOAP