SERVICE-ORIENTED ARCHITECTURE (CONTD..)

DR. MANJUNATH V HEGDE AND DR. VIDYA RAO

L19

MESSAGE-ORIENTED MIDDLEWARE

- MESSAGE-ORIENTED MIDDLEWARE (MOM) IS SOFTWARE OR HARDWARE INFRASTRUCTURE SUPPORTING SENDING AND RECEIVING MESSAGES BETWEEN DISTRIBUTED SYSTEMS.
- MOM ALLOWS APPLICATION MODULES TO BE DISTRIBUTED OVER HETEROGENEOUS PLATFORMS AND REDUCES THE COMPLEXITY OF DEVELOPING APPLICATIONS THAT SPAN MULTIPLE OPERATING SYSTEMS AND NETWORK PROTOCOLS.

MESSAGE-ORIENTED MIDDLEWARE

The study of MOM includes

- Enterprise Bus
- Publisher-Subscriber Model and Notification
- Queuing and Messaging Systems
- Cloud or Grid Middleware Applications

MESSAGE-ORIENTED MIDDLEWARE

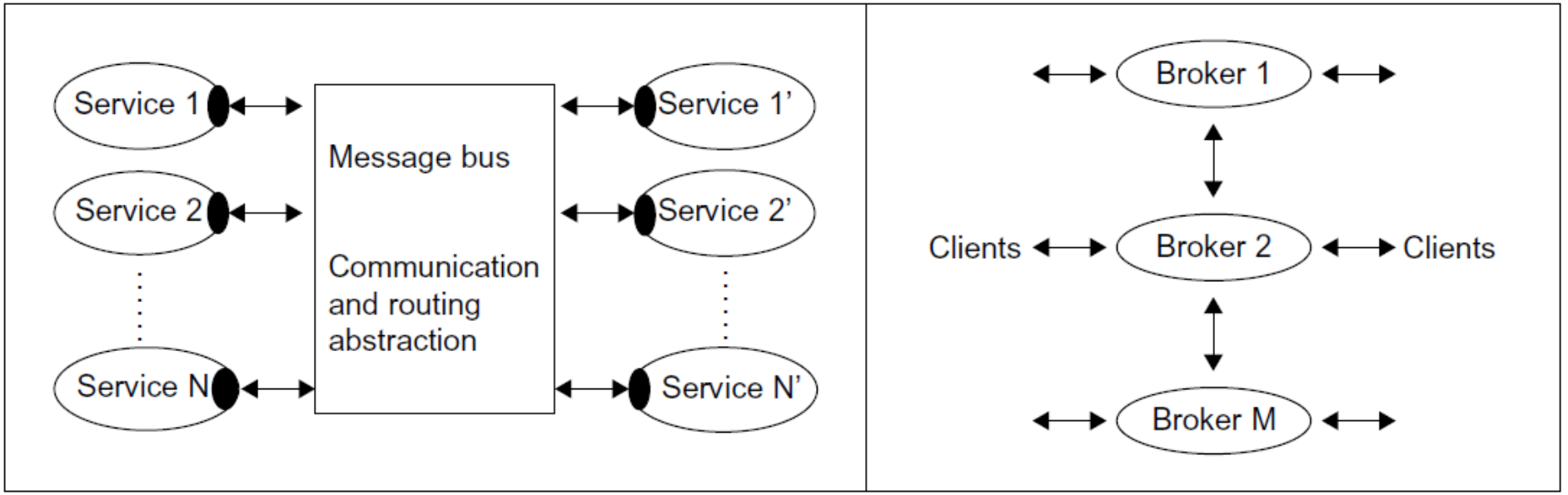
- **Message-oriented middleware (MOM)** is software or hardware infrastructure supporting **sending and receiving messages** between distributed systems.
- MOM allows application modules to be **distributed over heterogeneous platforms** and reduces the complexity of developing applications that span multiple operating systems and network protocols.
- The middleware **creates a distributed communications layer** that insulates the application developer from the details of the various operating systems and network interfaces.

https://en.wikipedia.org/wiki/Message-oriented_middleware

ENTERPRISE BUS

- Also called as **Enterprise Service Bus** (ESB)
- ESB is a centralized software component that performs integrations to backend systems like translations of data models, deep connectivity, routing, and requests.
- Developers can use a single protocol to ‘talk’ to the ESB and issue commands that direct interactions between services and leave it to the ESB to translate the commands, route the messages, and transform the data as required to get the commands executed.
- This enables developers to spend dramatically less time integrating and much more time configuring and improving their applications.
- Ultimately the challenges of maintaining, updating, and scaling a centralized ESB proved so overwhelming and expensive that the ESB often delayed the very productivity gains

ENTERPRISE BUS



(a) Implemented between services

(b) As a network of distributed brokers

FIGURE 5.6

Two message bus implementations between services or using a broker network.

ENTERPRISE BUS

- One may wish to introduce a wrapper so that services expecting messages in different styles (say, SOAP, REST, or Java RMI) can communicate with each other.
- One does not open a channel between source and destination, but rather injects a message into the bus with enough information to allow it to be delivered correctly. This injection is performed by code loaded into each service and represented by the filled ovals as client interfaces in Figure 5.6(a).
- The message bus is shown as linking services in the figure, but it can work with any software or hardware entity sending and receiving messages.

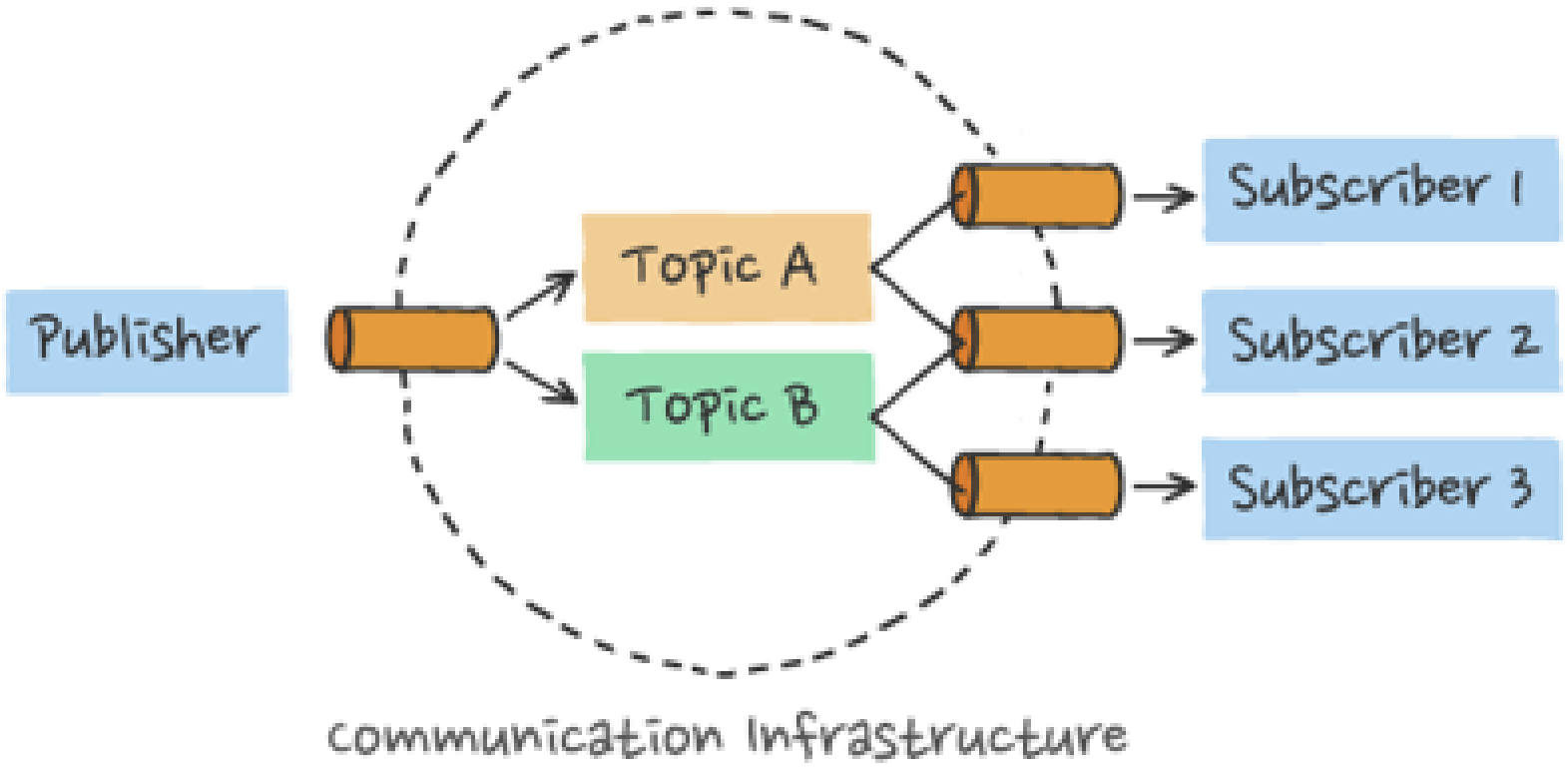
ENTERPRISE BUS

- These buses can be implemented internally to an application, or in a distributed fashion.
- i.e. the message bus typically can be implemented as a set of “brokers”.
- One implements brokers as managers of queues, and software in this area often has MQ or “Message Queue” in its description.

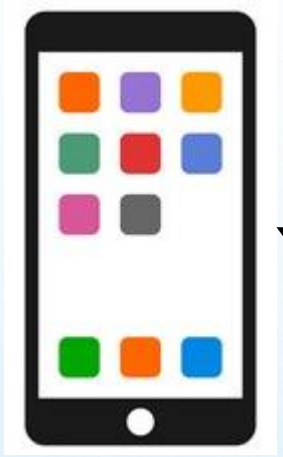
PUBLISHER-SUBSCRIBER MODEL AND NOTIFICATION

- A model for linking source and destination for a message bus.
- Here the producer of the message (publisher) labels the message in some fashion; often this is done by associating one or more topic names from a (controlled) vocabulary.
- Then the receivers of the message (subscriber) will specify the topics for which they wish to receive associated messages.
- The use of topic or content-based message selection is termed message filtering.

PUBLISHER-SUBSCRIBER MODEL AND NOTIFICATION

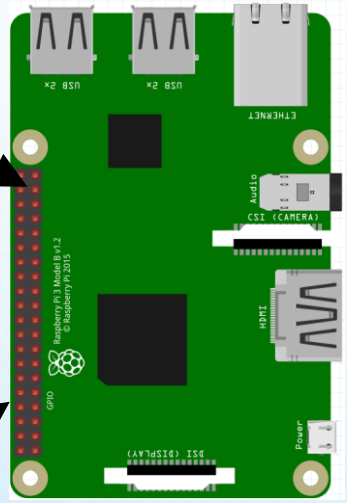


PUBLISHER-SUBSCRIBER MODEL AND NOTIFICATION



Subscriber-1

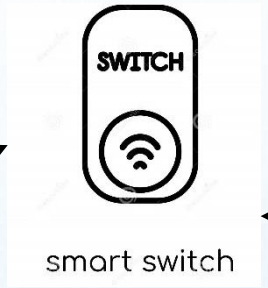
Topics:
Bulb
Fan



MQTT Broker

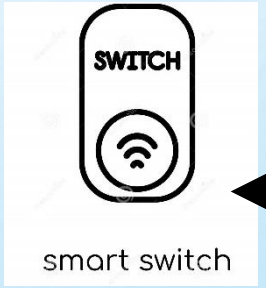
Bulb
Fan

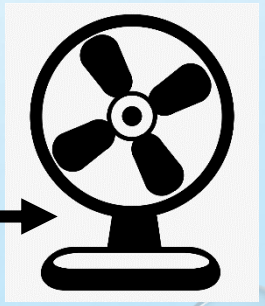
Bulb



Publisher-1

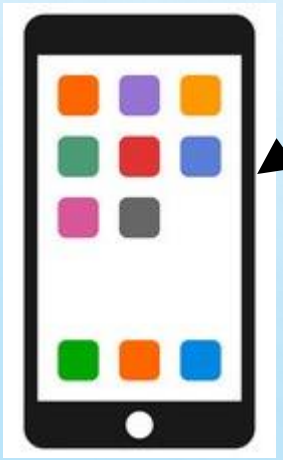
Fan





Publisher-2

Topics:
Bulb



Subscriber-2

QUEUING AND MESSAGING SYSTEMS

- **Java Message Service (JMS) -** specifies a set of interfaces outlining the communication semantics in pub/sub and queuing systems.
- **Advanced Message Queuing Protocol (AMQP) -** specifies the set of wire formats for communications; unlike APIs, wire formats are cross-platform.
- **MuleMQ, -** is a messaging framework underlying the ESB.
- The focus of Mule is to simplify the integration of existing systems developed using JMS, Web Services, SOAP, JDBC, and traditional HTTP.
- **Protocols supported within Mule** include POP, IMAP, FTP, RMI, SOAP, SSL, and SMTP

CLOUD OR GRID MIDDLEWARE APPLICATIONS

Environmental Monitoring and Internet Conference Using NaradaBrokering:

The GOAT project at Clemson University is part of the Program of Integrated Study for Coastal Environmental Sustainability (PISCES), which addresses environmental sustainability issues that can accompany coastal development. The current study incorporates groundwater monitoring, surface water quality and quantity monitoring, weather, and a variety of ecological measurements. The project utilizes the publishsubscribe messaging system, NaradaBrokering, to provide a flexible and reliable layer to move observation data from a wide variety of sensor sources to users that have diverse data management and processing requirements. NaradaBrokering can display environmental sensors.

CLOUD OR GRID MIDDLEWARE APPLICATIONS

Environmental Monitoring and Internet Conference Using NaradaBrokering: (contd..)

The commercial Internet Meeting software Anabas (www.anabas.com) incorporates support for sharing applications besides incorporating support for shared whiteboards and chat tools. Anabas uses NaradaBrokering for its content dissemination and messaging requirements. On a daily basis, Anabas supports several online meetings in the United States, Hong Kong, and mainland China. Note NaradaBrokering supports audio-video conferencing (using UDP) as well as other collaborative applications using TCP. Dynamic screen display published to NaradaBrokering can be displayed on collaborating clients.

CLOUD OR GRID MIDDLEWARE APPLICATIONS

QuakeSim Project for Earthquake Science using NaradaBrokering:

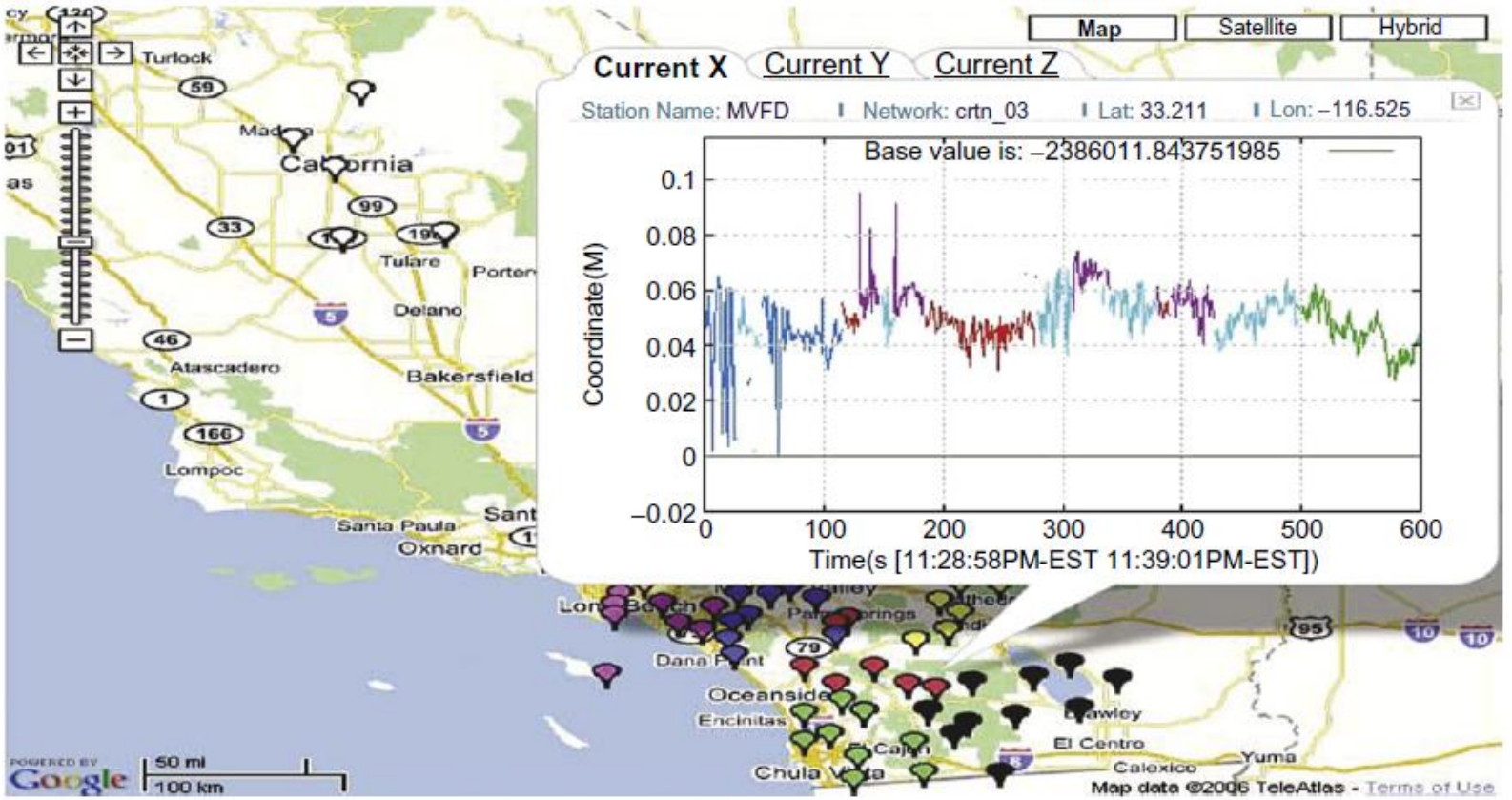


FIGURE 5.7

Display of GPS sensors managed by NaradaBrokering in Southern California; the map displays the time series produced by one of the GPS stations. (http://quakesim.jpl.nasa.gov/).

PORTALS AND SCIENCE GATEWAYS

- Science Gateway Exemplars
- HUBzero Platform for Scientific Collaboration
- Open Gateway Computing Environments (OGCE)

PORTALS AND SCIENCE GATEWAYS

Gateways provide user-centric environments for interacting with remote computational resources through user interfaces that are typically (but not exclusively) built with web technologies.

TurnKey Solution

HUBzero

Provide end-to-end Solution

Tool Box Solution

Open Gateway Computing Environment (OGCE)

Provides tools to solve a specific problem

PORTALS AND SCIENCE GATEWAYS

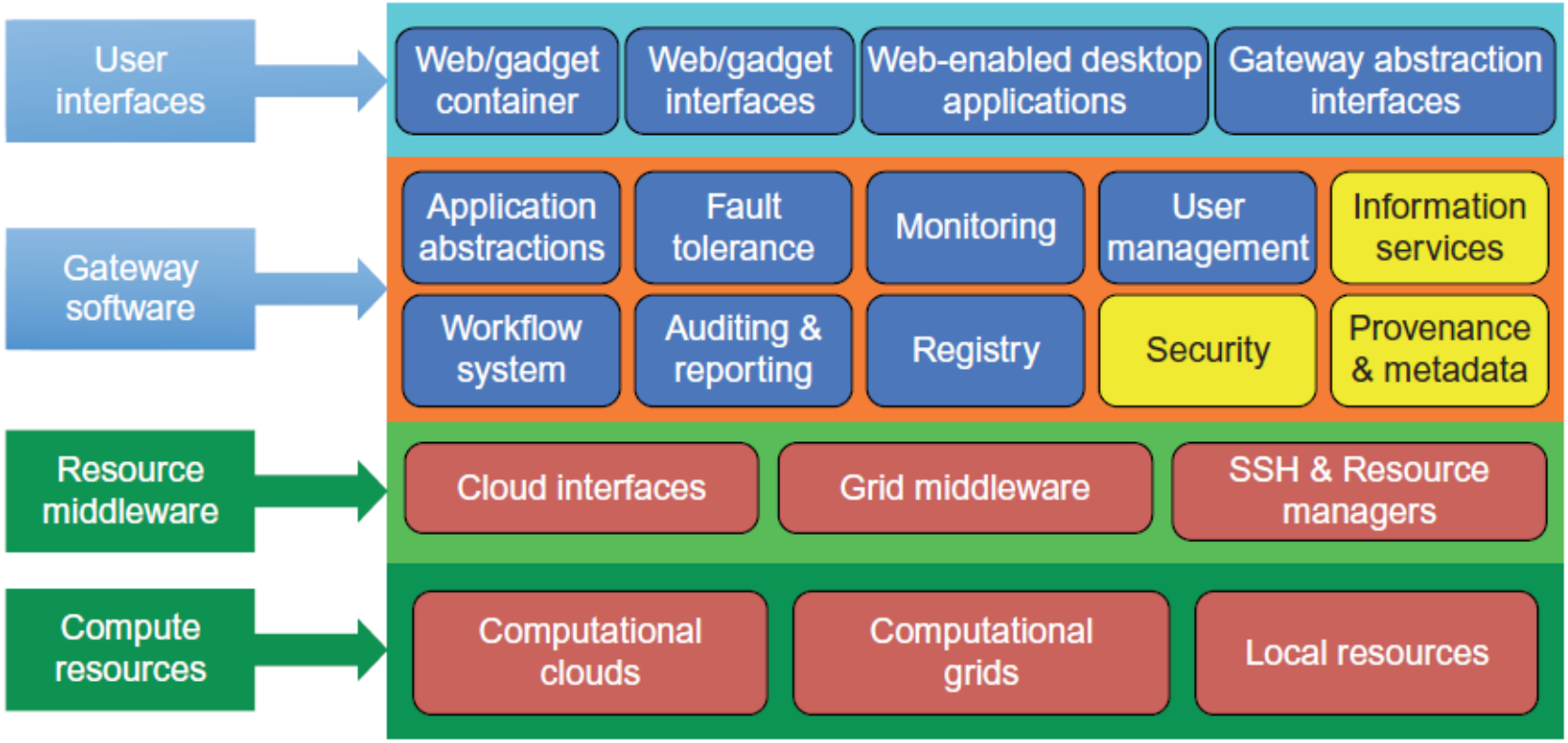


FIGURE 5.8

A gateway component software stack for scientific applications.

NEXT CLASS….

ABC