Module	Distributed Systems
Credits	5 ECTS / 3 U.S. semester credits
Important notes	Please note that this module is advanced and is intended for final year students only.
Allocation of marks	40% Continuous assessment 60% Final examination

Intended Module Learning Outcomes

On successful completion of this module learners will be able to:

1. explain challenges involved in designing and developing distributed systems
2. compare different architectural and communication modesl
3. write distributed applications using distributed object model and web services.
4. explain clock synchronisation and system state capturing strategies
5. evaluate appropriate distributed algorithms for leader election and mutual exclusion
6. describe state of the art distributed search techniques

Module Objectives

The aim of this module is to give the learner a clear overview of the challenges and design goals of distributed systems. Learners gain exposure to the issues that must be dealt with while constructing flexible distributed applications. Through emphasis on the conceptual basis, learners arealso taught different approaches to implement distributed applications.

Module Curriculum

Distributed systems overview

- Design goals and challenges: heterogeneity, openness, sccalability, consistency, transparency, mobility, fault tolerance, security
- Fundamental concepts: multicomputer vs multiprocessor systems
- tightly vs loosely coupled systems, horizontal vs vertical distribution
- Middleware based systems.
- Architectural models: client server model, multiple servers, thin clients, peer to peer model, mobile devices, spontaneous networking.

Interprocess communication

- Synchronous and Asynchronous communication
- TCP sockets
- Remote Procedure Calls
- message passing
- message oriented middleware

Distributed object model

- Distributed objects
- Object Request Broker
- object references
- Interfaces
- naming and binding
- method invocation
- Java RMI
- CORBA and Distributed Component Object Model
- Distributed object model design issues.

Service oriented architecture

- Web services
- XM:-RPC
- Web Services Description Language (WSDL)
- Simple Object Access Protocol (SOAP)
- Universal Description, Discovery and Integration (UDDI)
- Service oriented architecture: design issues.

Clock synchronisation

- Concept of time in distributed systems
- Physical vs logical time
- Clock synchronisation algorithms
- Network Time Protocol (NTP)
- Vector clocks

Distributed system states

- Global state predicates: deadlocks, termination detection, garbage collection
- Distributed snapshots: Chandy and Lamport snapshot algorithm, “possibility” and “definitely” evaluation.

Coordination, agreement and distributed search

- Distributed Algorithms: fairness, liveliness, safety
- Distributed Algorithms for: leader election, mutual exclusion and distributed search.