1: Introduction

General Context

The internet is very dynamic – devices connect and disconnect frequently, especially mobile devices.

The internet of things extends to all tiny communicating devices.

Most applications now are distributed.

Distributed Applications

Distributed applications are composed of several components that run as processes on different computers, communicating with one another across the network.

- You can also have a distributed application running on one host (e.g. smartphones, tablets)

Processes belonging to the same application may:

- Communicate with each other
- Share resources
- Invoke each other
- Synchronise their execution

Computing Models

1. Hard-coding: for the first distributed applications, code was deployed on fixed-address networked computers – all communication was hardcoded to known addresses.
2. Client/server: clients send requests to servers (with well known network and port addresses) that eventually return results.
    - raised level of abstraction – clients and servers are only aware of each other's names and attributes

3. Peer-to-peer: each peer can be client and/or server.

Features and Requirements of Distributed Systems

1. Heterogeneity at all layers (different hardware, OS, protocols may be used by different participating devices)
    - Requirement: Hide the heterogeneity by providing the means to overcome it / deal with it
2. Network has an impact on the performance and robustness of the application – can't guarantee performance
    - Requirement: Make the communication environment reliable and predictable
3. Need to link new software to legacy software (especially in industry)
    - Requirement: Provide interoperability

Challenges

- Hiding heterogeneity and complexity
- Managing dynamic environments efficiently
    - new services to be learned of, some services may move to different locations, etc.
    - examples of services: printing, camera, etc.
    - how do we describe new services to make clients aware of them?
        * need agreed ways of describing things
    - need a discovery service
- Getting predictable performance
    - generally this is costly
- Managing mobility and ad-hoc networks transparently
    - e.g. if you're using your phone walking down the street, you are connecting to different access points as you go – this requires a lot of work that is hidden from you

Middleware

- Middleware is the way to overcome these challenges
- common set of ubiquitous services that can be used by any distributed application in order to provide the following things:
  - transparency
  - interoperability
  - access to remote, dynamic services
  - mobility management
  - scalability
  - security
  - reduced response time
  - reliability
- not all distributed applications need middleware
  - though all use DNS, which is a typical middleware example

Location

- placed between applications and the OS
  - extends the set of services provided by the OS
  - applications access middleware services by their specific APIs

Middleware Examples

- Name services (DNS)
- Remote execution
- Discovery of services
- Event notification
  - between components of the application or of different applications
- Messaging
- Security and privacy
- Joining/leaving networks and mobility management

- Caching and content adaptation

Example 1

- live traffic webcams, video streams are fed to control centre, which can reorganise/direct traffic
- could also be public – different webcams can be discovered by potential users and users can get info
- directory service binds service names to attributes
  - attributes can be used to lookup service names

Example 2

- mobile users browse on the internet on the move
- device is changing access points, gets a new IP each time
- user is not aware of this, because all networking operations are hidden by the middleware

Middleware Implementations

- Remote Method Invocation (RMI)
- Distributed object systems (DCOM, CORBA)
- .NET
- Enterprise middleware (TPM, MOM)
- Cluster, grid, and cloud management systems (Globus, Amazon EC2, Microsoft Azure)