

# Announcements

R3 - There will be Presentations

Clarify any requirements and expectations with stakeholder

Unify any assumptions/dependencies with other silos



# Distributed Systems

SWEN-343



# Distributed Systems - A Large Topic

Frequently an entire course

We will be focusing on it for one day

Focusing on things from an Enterprise Perspective  
Especially Reliability



Imagine we were creating an ERP system for  
1->n Users

What would some concerns be?

Functional quality

Data Management/Integrity

Security

“Update Management”

.. Many More.....



# Why Not Just One Machine?

Probably not a realistic option

Too much weight

Not possible

Different machines have different purposes

What if 1 machine goes down?



# Distributed Systems - Definition

“A distributed system is a collection of independent computers that appear to the users of the system as a single computer.”



# Distributed System Characteristics

**Fault-Tolerant:** Recover from component failures without performing incorrect actions.

**Highly Available:** Restore operations, permitting it to resume providing services even when some components have failed.

**Recoverable:** Failed components can restart themselves and rejoin the system, after the cause of failure has been repaired.

**Consistent:** Coordinate actions by multiple components often in the presence of concurrency and failure.



# Transactions (flashback)

- Bounded sequence of work with start and end points
- Resources are in consistent state throughout
- Completion on all-or-nothing basis
- ACID properties as discussed previously:
  - Atomicity – bound and completed
  - Consistency – non-corrupt state
  - Isolation – results only visible until transaction “commits successfully”
  - Durability – permanent if committed “survives crash”.





# Distributed System Characteristics

**Scalable:** It can operate correctly even as some aspect of the system is scaled to a larger size.

**Predictable Performance:** The ability to provide desired responsiveness in a timely manner.

**Secure:** The system authenticates access to data and services



# Other - Challenges

Load Balancing

Quality Assurance

Releasing concurrent updates

Ensuring a Maintainable System

... Others.....



# Distributed Systems - Handling Failure

What happens if a node/component goes down?

How do you ensure that all data is successfully transmitted?

How to ensure that the correct data is transmitted?

Will address all in activity



# Architecture & Design

Important for an ERP & Important in a  
Distributed System



# Architecture & Design - Challenges

Multiple platforms, languages

Proprietary, legacy systems

Conflicting data formats

Integration not planned for in original design



# Architecture & Design - Possible Solutions

Middleware, design patterns, & frameworks

Facade

Adapter

Decorator

Observer

Proxy

... Others.....

Standards based development

Component based development



# Architecture & Design - Uniform Software

Uniform access to services

Uniform discovery of resources and object names

Uniform error handling methods

Uniform security policies



# Recap

Distributed systems are necessary

- Load balancing

- Redundancy

## Challenges

- Fault tolerance

- Maintainability

- Scalability

## Solutions: Design

- Patterns

- Component based development

- No silver bullet. Be a good SE.





# Questions

