

Purpose

The system design is documented in the System Design Document (SDD). It describes additional design goals set by the software architect, the subsystem decomposition (with UML class diagrams), hardware/software mapping (with UML deployment diagrams), data management, access control, control flow mechanisms, and boundary conditions. The SDD serves as the binding reference document when architecture-level decisions need to be revisited.

Audience

The audience for the SDD includes the system architect and the object designers as well as the project manager.

Table of Contents

1 Subsystem Decomposition	2
2 Hardware/Software Mapping	2
3 Persistent Data Management	2
4 Boundary Conditions	2

Document History

Rev.	Author	Date	Changes
1	name	31st April 2022	Sample changes 1
2	name	15th May 2022	Sample changes 1
3	name	29th May 2022	Sample changes 1
4	name	12th June 2022	Sample changes 1
5	name	26th June 2022	Sample changes 1

1 Subsystem Decomposition

This section describes the decomposition of the system into subsystems and the services provided by each subsystem. The services are the seed for the APIs detailed in the Object Design Document.

2 Hardware/Software Mapping

This section describes how the subsystems are mapped onto existing hardware and software components. A UML deployment diagram accompanies the description. The existing components are often off-the-shelf components. If the components are distributed on different nodes, the network infrastructure and the protocols are also described.

3 Persistent Data Management

This section describes how the entity objects are mapped to persistent storage. It contains ction-Global Software Control This section describes the control flow of the system, in particular, whether a monolithic, event-driven control flow or concurrent processes have been selected, how requests are initiated and specific synchronisation issues.

4 Boundary Conditions

This section describes the use cases how to start up the separate components of the system, how to shut them down, and what to do if a component or the system fails.