SmartHomeEnergy SoSD

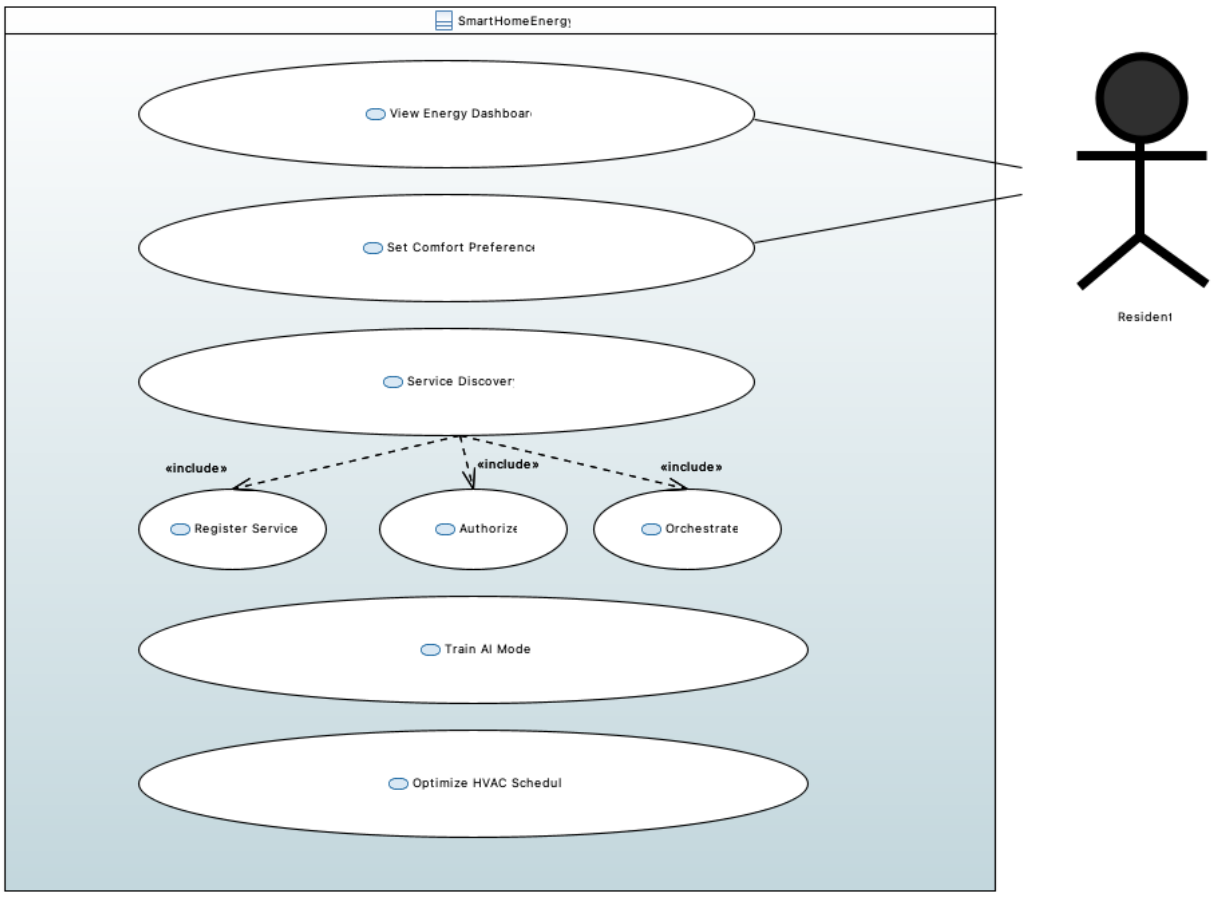
System Description

Abstract
This is the template for System of Systems Description (SoSD document) according to the Eclipse Arrowehad documentation structure.

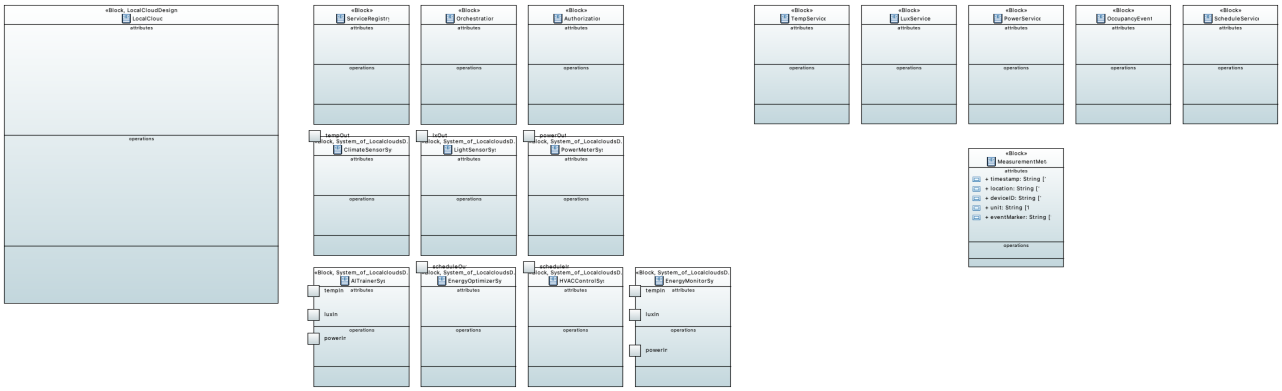
Contents

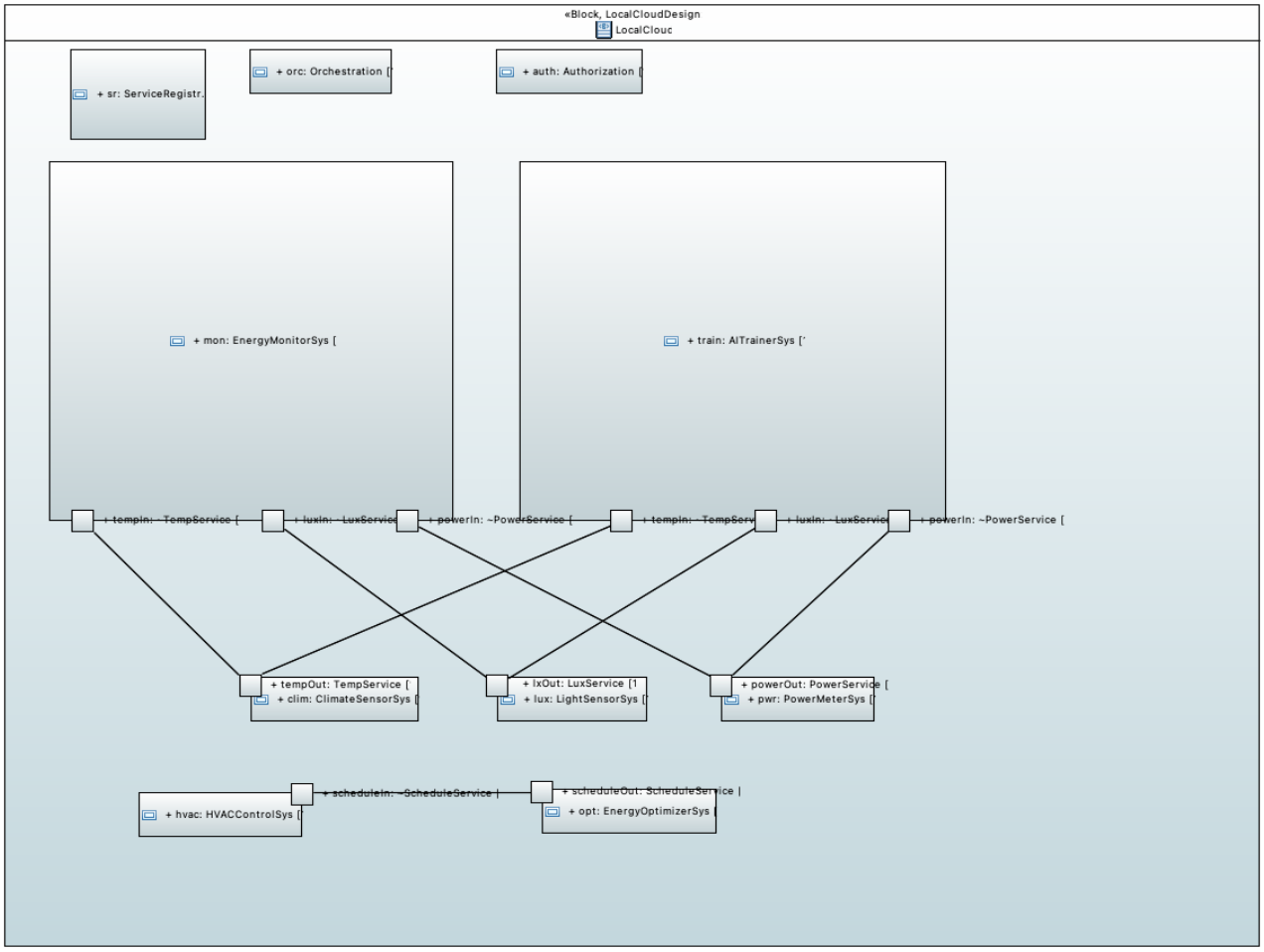
1 Overview

SmartHomeEnergy reduces household energy use while maintaining comfort. It executes inside a single Local Cloud and uses Arrowhead core systems — Service Registry, Orchestration, and Authorization — for discovery and secure late binding at runtime.



The SoS hosts sensors (temperature, light, power), AI training, an energy optimizer that produces an HVAC schedule, and a controller that applies that schedule. The Resident actor views a dashboard and sets comfort preferences.




«Block, LocalCloudDesign
LocalClouc
+ sr: ServiceRegistr.
+ orc: Orchestration [
+ auth: Authorization [
+ mon: EnergyMonitorSys [
+ train: AITrainerSys ['
+ tempIn: ~TempService [
+ luxIn: ~LuxService
+ powerIn: ~PowerService [
+ tempIn: ~TempServ
+ luxIn: ~LuxServic
+ powerIn: ~PowerService [
+ tempOut: TempService [
+ clim: ClimateSensorSys [
+ lxOut: LuxService [1
+ lux: LightSensorSys [
+ powerOut: PowerService [
+ pwr: PowerMeterSys [
+ scheduleIn: ~ScheduleService |
+ scheduleOut: ScheduleService |
+ hvac: HVACControlSys [
+ opt: EnergyOptimizerSys

1.1 Significant Prior Art

Omitted for beginner scope (not required for simple services).

1.2 How This SoS Is Meant to Be Used

Deployed on a home gateway (Local Cloud). Sensors publish Temp/Lux/Power; **AITrainerSys** builds a model; **EnergyOptimizerSys** outputs an HVAC schedule consumed by **HVACControlSys**. Resident uses the dashboard and sets preferences.

SoS functionalities and properties

Functional

- F1 – Collect Temp/Lux/Power measurements.
- F2 – Train model on historical measurements.
- F3 – Produce optimized **ScheduleService** for HVAC.
- F4 – Visualize measurements and consumption.

Non-functional

- N1 – Security: service consumption requires Authorization *permit*.
- N2 – Loose coupling: discovery + late binding via Orchestration.
- N3 – Resilience: if data is missing, fall back to last valid schedule.

Stateful or stateless

Stores historical samples and the last optimized schedule.

1.3 Important Delimitations

No protocols, encodings, or deployments are specified (white-box/IDD/implementation out of scope at beginner level).

2 Services

2.1 Produced service

Producer System	Service	Brief purpose	SD
EnergyOptimizerSys	ScheduleService	Optimized HVAC schedule (start/stop times, setpoints)	SD_ScheduleService

2.2 Consumed services

Consumer	Service	Provider	Purpose	SD
AITrainerSys	TempService	ClimateSensorSys	Model input (temperature)	SD_TempService
AITrainerSys	LuxService	LightSensorSys	Model input (illuminance)	SD_LuxService
AITrainerSys	PowerService	PowerMeterSys	Model input (power usage)	SD_PowerService
EnergyMonitorSys	TempService/ LuxService/ PowerService	Sensors	Dashboard visualization	
HVACControlSys	ScheduleService	EnergyOptimizerSys	Apply optimized schedule	SD_ScheduleService

3 Security

Arrowhead-style security is assumed within the Local Cloud.

- **Identity**: Every system in the Local Cloud has an Arrowhead-compliant X.509 certificate.
- **Discovery**: Consumers discover providers via *Service Registry*; no hard-coded endpoints.
- **Authorization**: Before a provider endpoint is returned, *Orchestration* consults *Authorization*. Only permitted consumer–service pairs are allowed.
- **Confidentiality & Integrity**: Service invocations use TLS within the Local Cloud trust boundary.
- **Least privilege**: Policies are per service; only needed services are permitted for each consumer.

Security model

Secure late binding at runtime:

1. Consumer requests service (e.g., `TempService`) from *Orchestration*.
2. *Orchestration* calls *Authorization*; if *permit*, it returns provider endpoint(s).
3. Consumer calls provider over TLS using its certificate.

4 Revision History

4.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	2025-10-13	1.0.0	Initial SoSD (beginner scope)	Abdalrahman Nasser