POWER2DM

"Predictive model-based decision support for diabetes patient empowerment"

Research and Innovation Project
PHC 28 – 2015: Self-management of health and disease and decision support systems based on predictive computer modelling used by the patient him or herself

POWER2DM D1.4

Software Test Plan for POWER2DM Infrastructure and its Components (D1.4.1)

Due Date:	30th November 2016 (M10)
Actual Submission Date:	13th January 2017, resubmitted on 20 April 2017
Project Dates:	Project Start Date: February 01, 2016 Project End Date: July 31, 2019 Project Duration: 42 months
Deliverable Leader:	PD

Project co-funded by the European Commission within H2020 Programme (20015-2016)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document History:

Version	Date	Changes	From	Review
V0.1	01-11-2016	Initial document	PD	
V0.2	02-12-2016	Updated version	PD	
V0.3	14-02-2016	Updated version	PD	TNO, SRFG, SRDC
V0.4	12-01-2017	Updated version	PD	TNO
V1.0	18-01-2017	Updated version	PD	European Commision
V1.1	15-03-2017	Updated version based on review comments	PD	TNO
V1.2	20-04-2017	Updated version based on review comments	PD	

Contributors (Benef.) Marlies Schijf (PD)
Roosmarijn Schopman (PD)
Albert De Graaf (TNO)
Tuncay Namli (SRDC)

Responsible Author Marlies Schijf **Email** m.schijf@primedata.nl

POWER2DM CONSORTIUM PARTNERS

Abbv	Participant Organization Name	Country
TNO	Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek	Netherlands
IDK	Institute of Diabetes "Gerhardt Katsch" Karlsburg	Germany
SRDC	SRDC Yazilim Arastirma ve Gelistirme ve Danismanlik Ticaret Limited Sirketi	Turkey
LUMC	Leiden University Medical Center	Netherlands
SAS	SAS Servicio Andaluz de Salud	Spain
SRFG	Salzburg Research Forschungs Gesellschaft	Austria
PD	Prime Data B.V.	Netherlands
iHealth	iHealth EU	France

ROADMAP

Doc. Version	Due Date	Expected/Planned Actions	From
V0.2	M10	Publish the deliverable (DEADLINE) D1.4 (D1.4.1 Software Test Plan for POWER2DM Infrastructure and its Components). This is a draft version that needs to be updated in M12.	PD, TNO, SRDC
V0.4	M12	Update of deliverable D1.4 (D1.4.1 Software Test Plan for POWER2DM Infrastructure and its Components). This is a draft version that needs to be updated in M13 with a finalized Product Risk Analysis and Test Strategy for the first prototype.	PD, TNO, SRDC
V1.0	M13	Finalisation of D1.4 (D1.4.1 Software Test Plan for POWER2DM Infrastructure and its Components) as a baseline for the first prototype and subsequent pilot. Review will be carried out beforehand by all partners. Final version	ALL
V2.0	M?	Possible update of D1.4 (D1.4.1 Software Test Plan for POWER2DM Infrastructure and its Components) dependent on scope changes for future prototypes. This document will be a reference to a DEM: Demonstrator, pilot, prototype. It is part of Prototype and System Release 1.	PD

OPEN ISSUES

No:	Date	Issue	Resolved
1	14.12.2016	Document is not reviewed yet by partners other than TNO and SRDC. Due to delay of the Quantification Campaign the scope of the first prototype is not clear yet at the moment of the deadline of this deliverable.	**M12**
2	12.01.2017	Definitions and Acronyms have not been listed yet.	**M13**
3	12.01.2017	This document can only be completed after scope of prototype 1 is known. An update will follow mid-January 2017	**M13**

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 PURPOSE

The purpose of this document is to describe the testing framework for the POWER2DM system. This testing framework will be used during every prototype development.
Throughout the project there are three release phases. During these release phases four prototypes will be released. The three release periods are:

- Release phase 1: release first release candidate
- Release phase 2: roadmap for future releases (release calendar)
- Release phase 3: release of new prototypes & management and maintenance of system

The testing framework details the guidelines for the process of development and testing at all software developing partners. After components are completed integration takes place, for which the process and planning are also described. During integration, several tests will be carried out to ensure correct connection and transfer and handling of data. Moreover, the requirements for privacy and security will be tested as well as performance.

Responsibilities for test activities depend on:

- What is tested.
- How will it be tested, for instance automated testing, test users, etc.
- The pre-requisites for testing, for instance dummy data, configurations, etc.

By using the testing framework, it is ensured that prototypes are ready for the pilot studies.

2 REFERENCE DOCUMENTS

The following documents were used or referenced in the development of this document:

- POWER2DM D1.3 Conceptual Design
- POWER2DM D4.1 Personal Data Model and Service API
- Devices used in the POWER2DM Quantification Campaign
  - This deliverable is used instead of the results of the delayed Quantification Campaign
- C.1 Power2DM Quantification Campaign Protocol LUMC v1_9 2016-07-21
  - This deliverable is used instead of the results of the delayed Quantification Campaign

2.1 Definitions and Acronyms

Table 1 List of Abbreviations and Acronyms

Abbreviation/ Acronym	DEFINITION
A	Test type: Automated test
API	Application Programming Interface
D	Deliverable (e.g. D1.4)
FAT	Test type: Factory Acceptance Test
I	Test type: Implicit test
JITAI	Just In Time Adaptive Intervention. JITAIs are self-management interventions that are initiated by the POWER2DM SMSS system (specifically recommender engine) automatically during daily life of the patient according to the changing context of the patient.
MTP	Master Test Plan
PDS	POWER2DM Personal Data Store component
PRA	Product Risk Analysis
R	Test type: Review
SMSS	POWER2DM Self-Management Support System
SAT	Test type: Site Acceptance Test
ST	Test type: System Test
UI	User Interface
UT	Test type: Unit Test

3 SYSTEM AND SCOPE

This section contains an overview of the integrated POWER2DM system, its subsystems, and the interfaces between the subsystems. For each subsystem and interface is indicated which party or parties are responsible for testing, in order to clarify the test scope per party.

3.1 Overview

The components that will be developed for the system are shown in Figure 1 below. They are:

- UI
  - User & privacy management web interface
  - Shared decision making web interface
  - SMSS mobile application
  - SMSS web interface
- Core
  - Consent management service
  - Identity service
  - Authorization service
- Personal Data Store
  - Personal data service
  - Passive sensor logging
  - Web/application experience logging service
- Prediction Services
  - KADIS prediction / simulation service
  - MT2D-MARVEL prediction / simulation service
  - Risk score calculation service
- Communication Engine
  - JITAI plan queue service
- Action Plan Engine
  - Action plan service

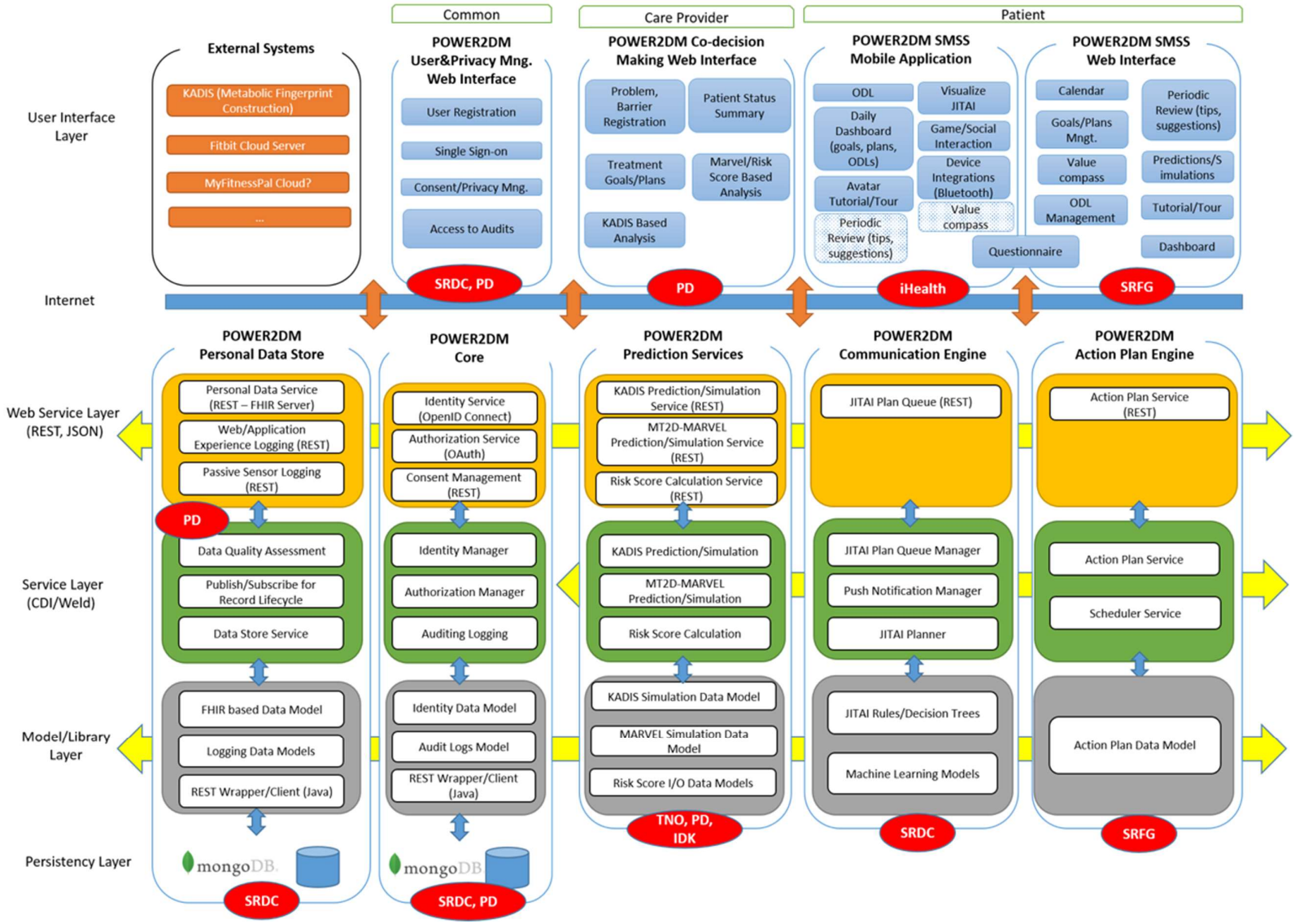

Figure 1, POWER2DM Conceptual Design

3.2 Subsystems and Interfaces

Based on the conceptual design the following system diagrams – Figure 2 till Figure 7 – showing the interfaces between the software subsystems can be determined:

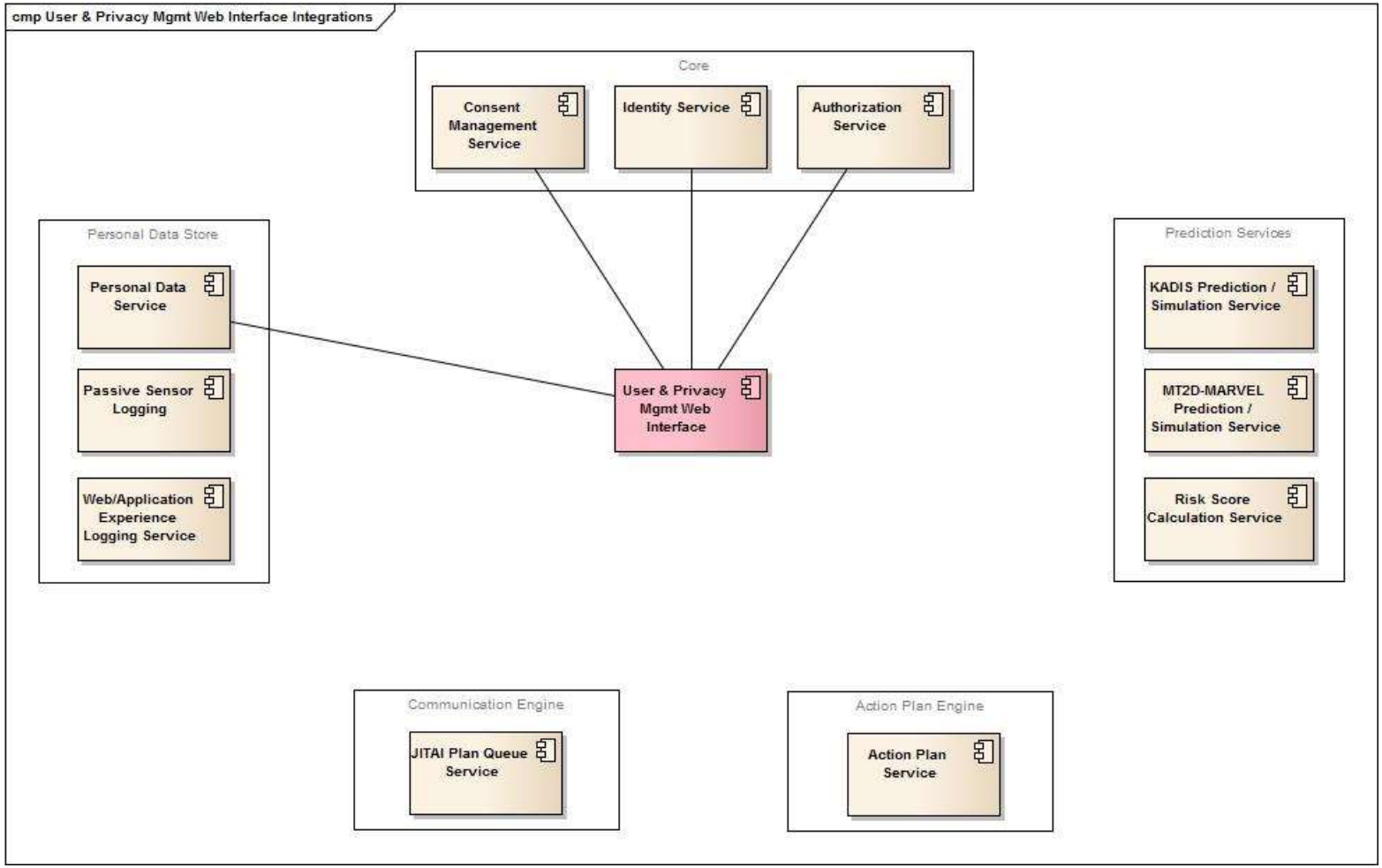

Figure 2, Interfaces User & Privacy management web interface

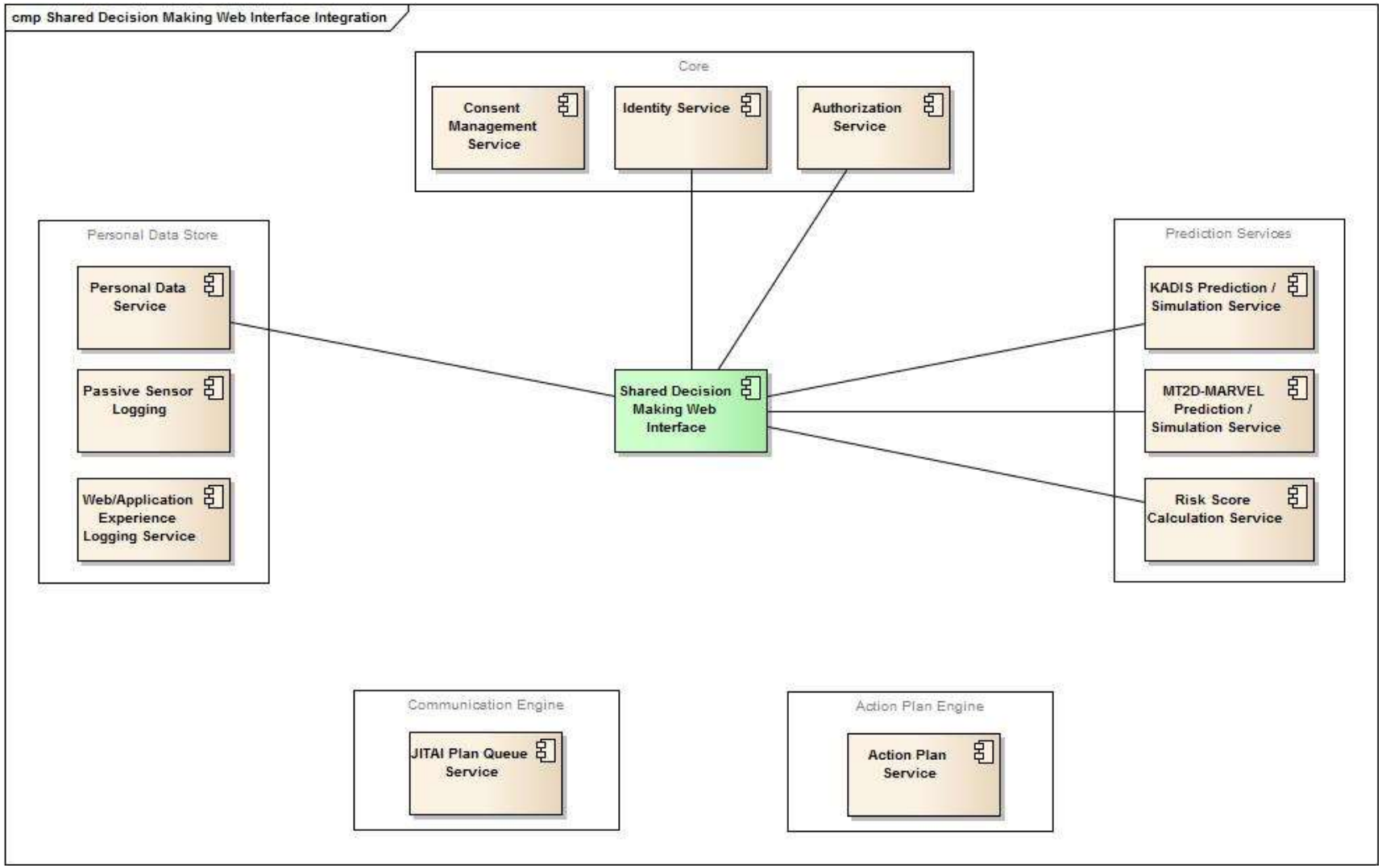

Figure 3, Interfaces Shared Decision Making web interface

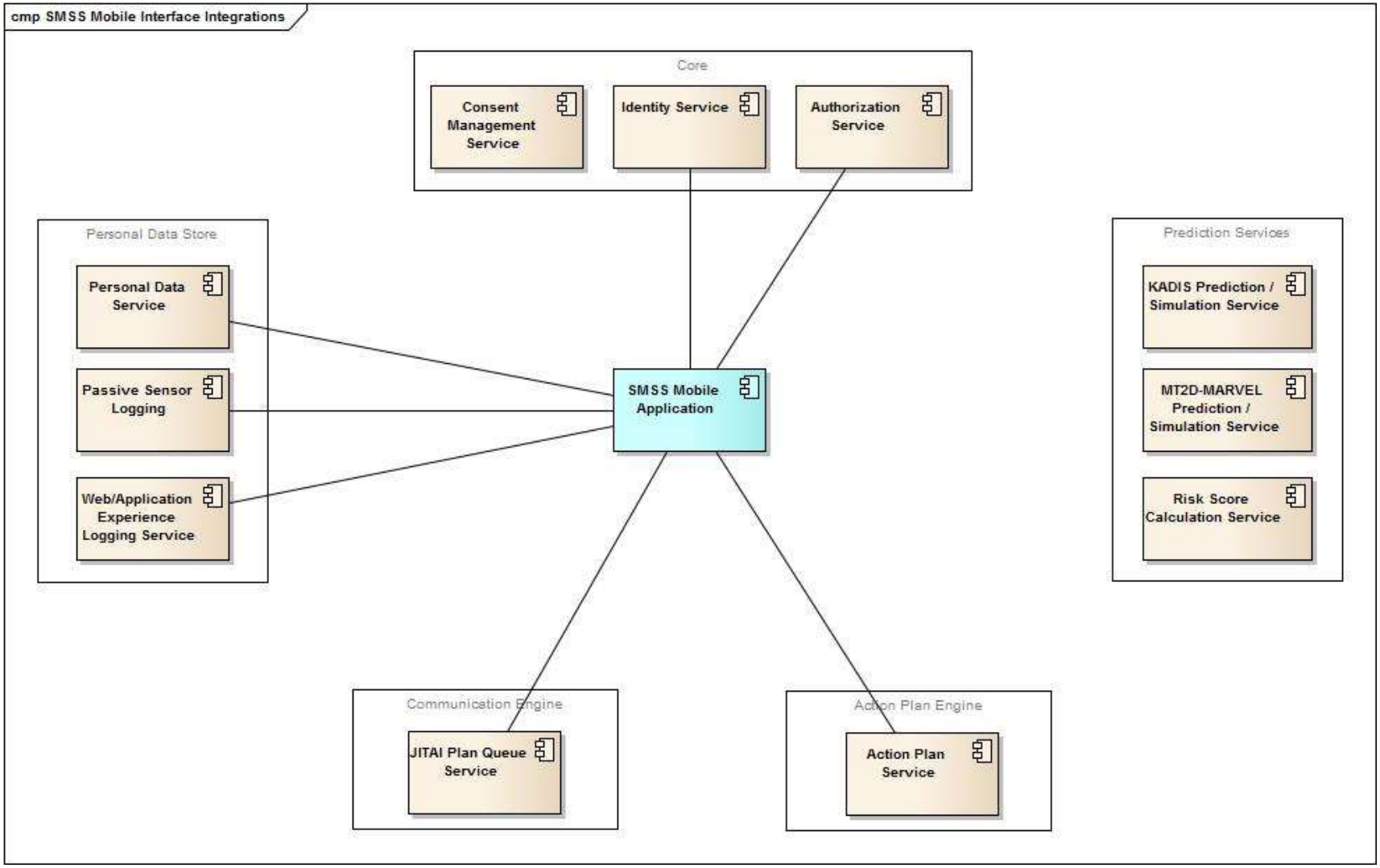

Figure 4, Interfaces SMSS mobile application

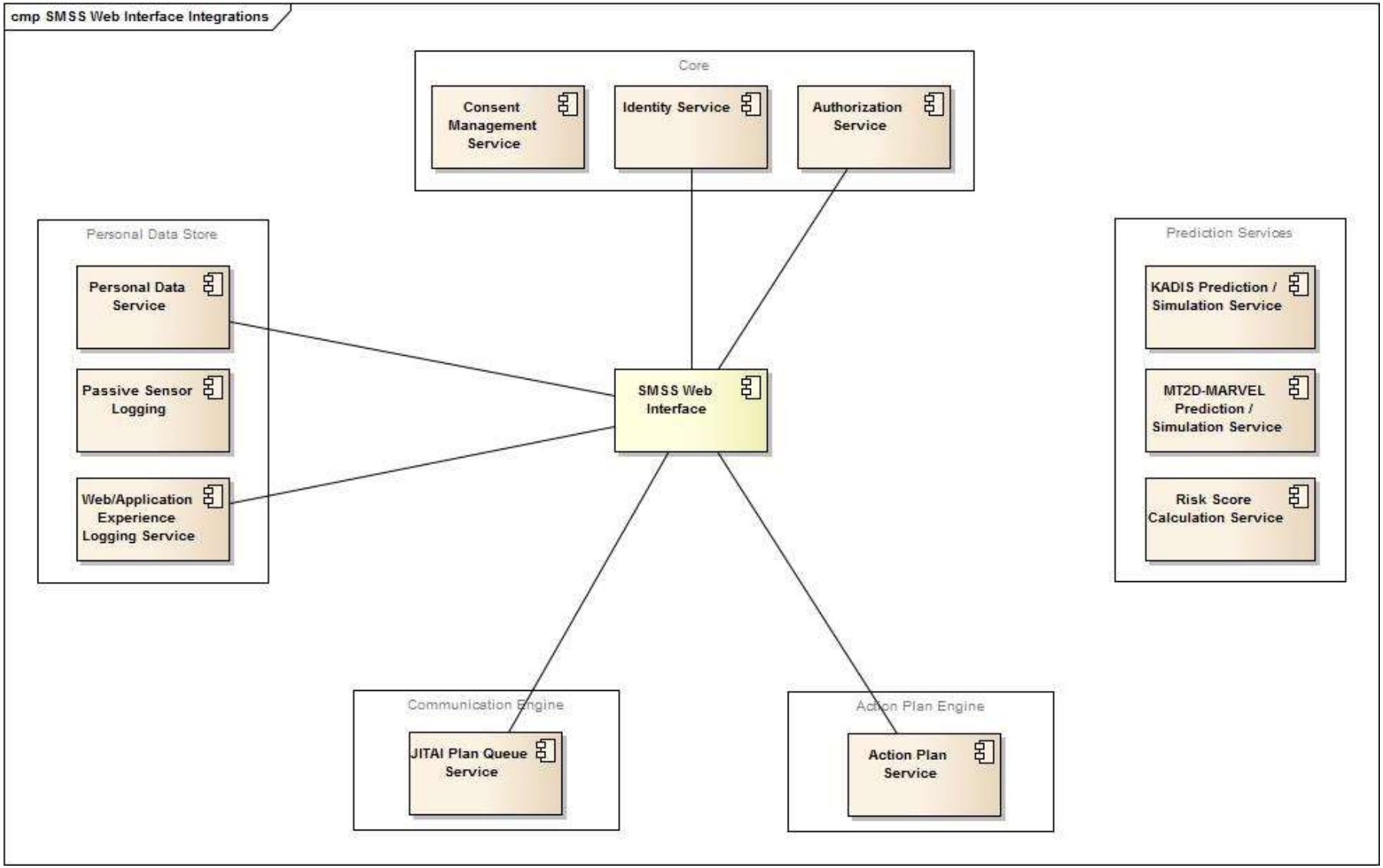

Figure 5, Interfaces SMSS web interface

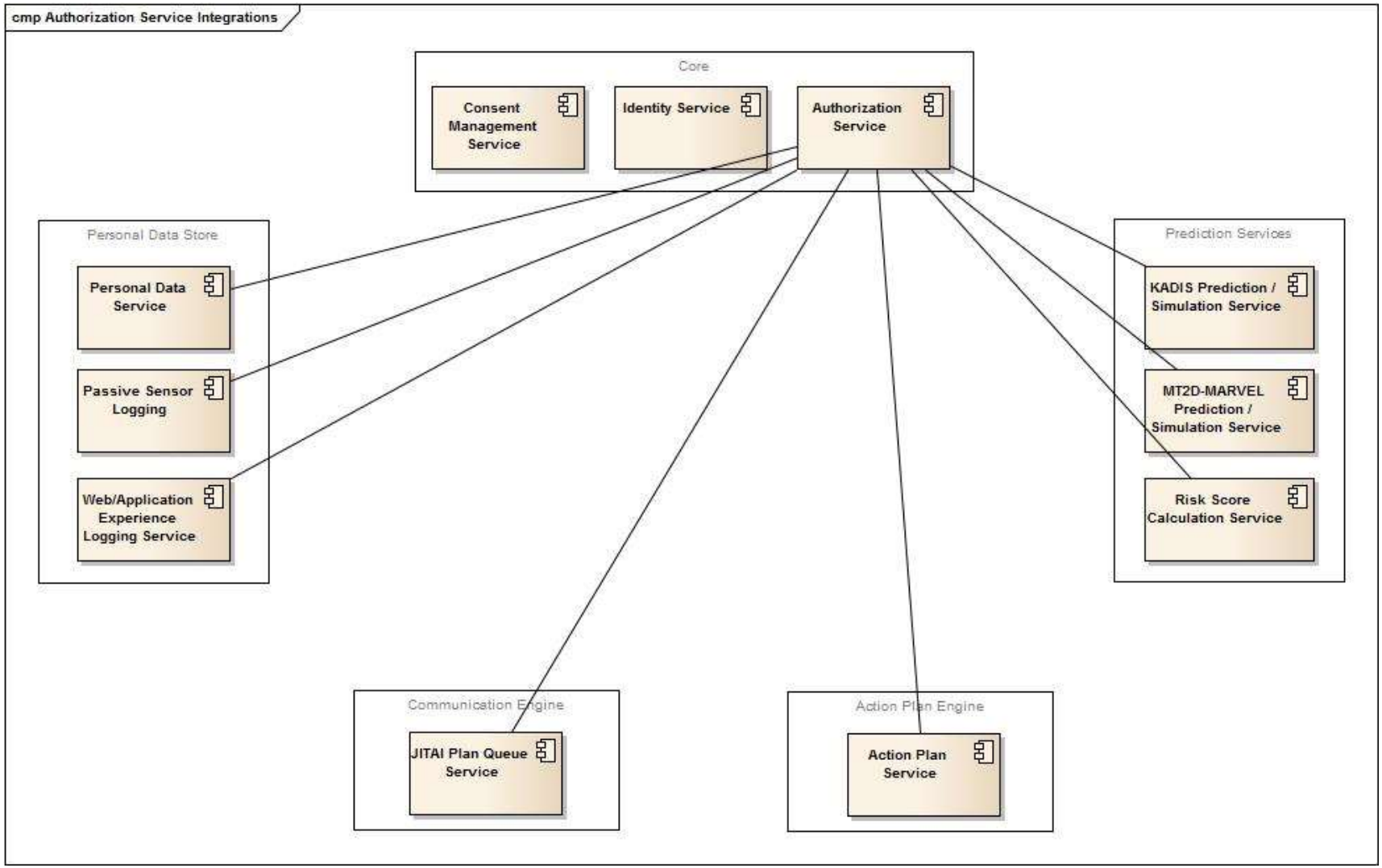

Figure 6, Interfaces Authorization service

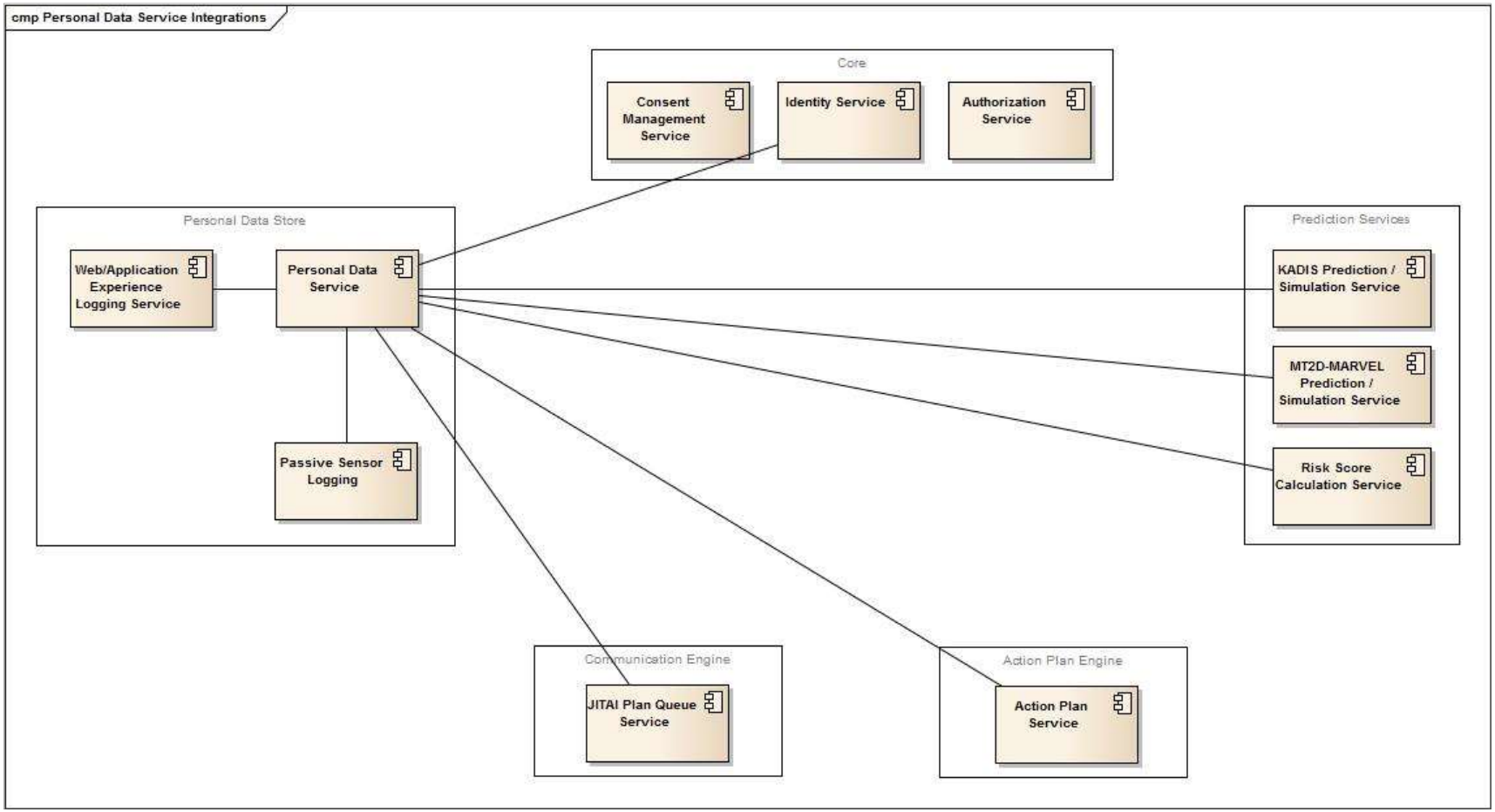

Figure 7, Interfaces Personal Data service

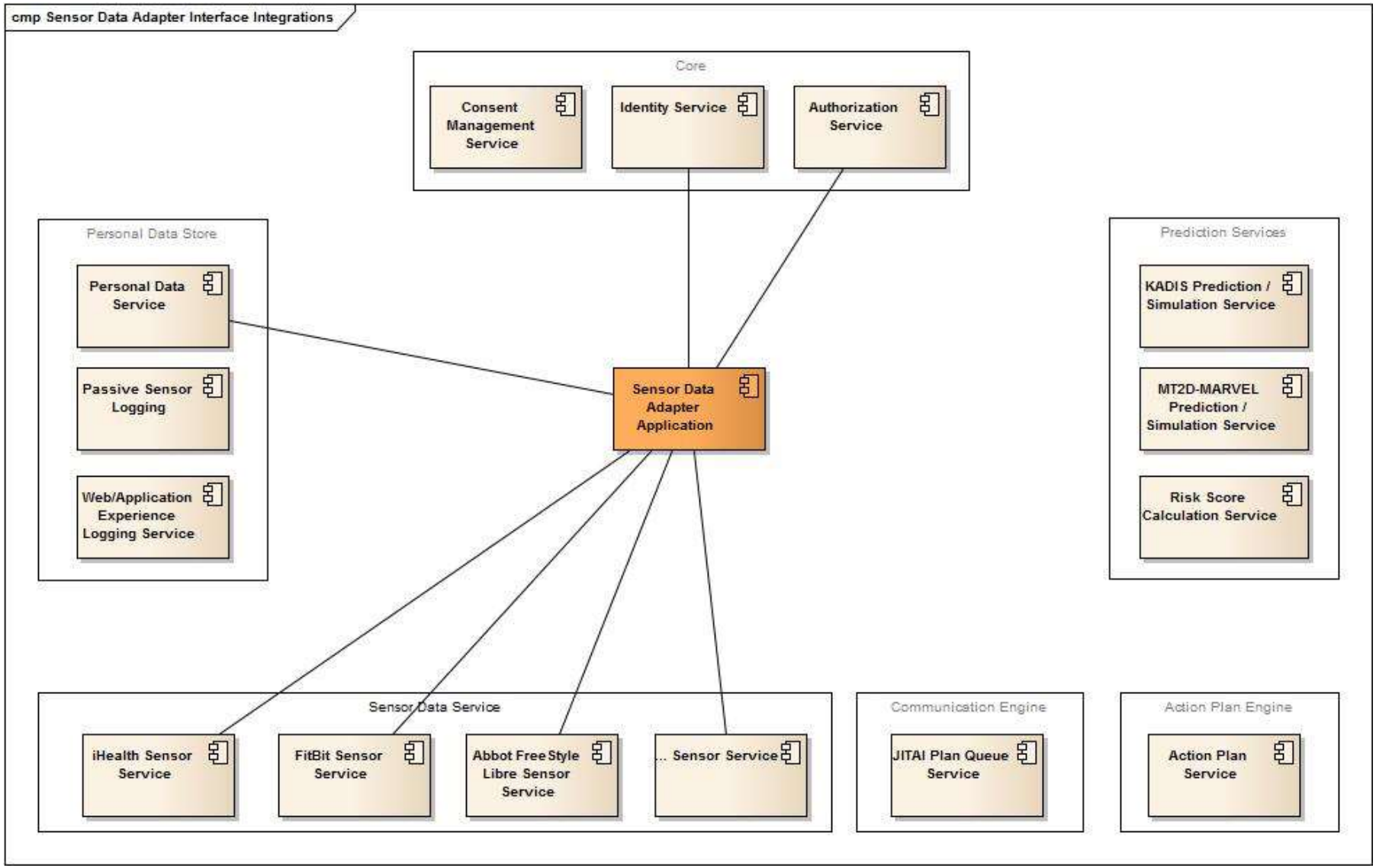

Figure 8, Interfaces Sensor Data Adapter application

Based on the conceptual design, the following table shows which party is responsible for which subsystem. Together with the responsibility of the development of a subsystem, comes the responsibility of performing sufficient system tests to for that subsystem. This includes the responsibility of arranging a representative test environment, test data and tools to perform the tests.

Table 2: Subsystems and responsible parties

Subsystem	Responsible party/parties
User & privacy management web interface	SRDC, PD
Shared decision making web interface	PD
SMSS mobile application	iHealth
SMSS web interface	SRFG
Consent management service	SRDC, PD
Identity service	SRDC, PD
Authorization service	SRDC, PD
Personal data service	SRDC, PD
Passive sensor logging	SRDC
Web/application experience logging service	SRDC
KADIS prediction / simulation service	IDK
MT2D-MARVEL prediction / simulation service	TNO
Risk score calculation service	TNO
JITAI plan queue service	SRDC
Action plan service	SRFG
Sensor data adapter application	PD, TNO

Based on these responsibilities and the interfaces between the relevant components, the following table gives an overview of the interfaces in the system and the parties responsible for testing them.

Table 3: Interfaces and responsible parties

Interface	Responsible parties
User & privacy management web interface – Consent management service	SRDC, PD
User & privacy management web interface – Identity service	SRDC, PD
User & privacy management web interface – Authorization service	SRDC, PD
User & privacy management web interface – Personal data service	SRDC, PD
Shared decision making web interface – Identity service	PD
Shared decision making web interface – Authorization service	PD
Shared decision making web interface – Personal data service	PD
Shared decision making web interface – KADIS prediction / simulation service	PD
Shared decision making web interface – MT2D-MARVEL prediction / simulation service	PD
Shared decision making web interface – Risk score calculation service	PD
SMSS mobile application – Identity service	iHealth
SMSS mobile application – Authorization service	iHealth
SMSS mobile application – Personal data service	iHealth
SMSS mobile application – Passive sensor logging	iHealth
SMSS mobile application – Web/application experience logging service	iHealth
SMSS mobile application – JITAI plan queue service	iHealth
SMSS mobile application – Action plan service	iHealth
SMSS web interface – Identity service	SRFG
SMSS web interface – Authorization service	SRFG
SMSS web interface – Personal data service	SRFG

SMSS web interface – Web/application experience logging service	SRFG
SMSS web interface – JITAI plan queue service	SRFG
SMSS web interface – Action plan service	SRFG
Personal data service – Identity service	SRDC, PD
Personal data service – Passive sensor logging	SRDC, PD
Personal data service – Web/application experience logging service	SRDC, PD
Personal data service – KADIS prediction / simulation service	SRDC, PD
Personal data service – MT2D-MARVEL prediction / simulation service	SRDC, PD
Personal data service – Risk score calculation service	SRDC, PD
Personal data service – JITAI plan queue service	SRDC, PD
Personal data service – Action plan service	SRDC, PD
Authorization service – Personal data service	SRDC, PD
Authorization service – Passive sensor logging	SRDC, PD
Authorization service – Web/application experience logging service	SRDC, PD
Authorization service – KADIS prediction / simulation service	SRDC, PD
Authorization service – MT2D-MARVEL prediction / simulation service	SRDC, PD
Authorization service – Risk score calculation service	SRDC, PD
Authorization service – JITAI plan queue service	SRDC, PD
Authorization service – Action plan service	SRDC, PD
Sensor data adapter – Identity service	TNO, PD
Sensor data adapter – Authorization service	TNO, PD
Sensor data adapter – Personal data service	TNO, PD
Sensor data adapter – iHealth sensor service	TNO, PD
Sensor data adapter – FitBit sensor service	TNO, PD
Sensor data adapter – Abbot FreeStyle Libre sensor service	TNO, PD

4 TEST STRATEGY

This Master Test Plan will mainly deal with the test approach for the system tests that have to be done during the development of the POWER2DM system.

The system tests, including system integration tests, will be concluded with a Factory Acceptance Test (FAT) before delivery of the software for the pilot sessions. This FAT will be performed for each of the components by the responsible partner. The integration of all components will be tested with PD in the lead. Fulfilling the software requirements and meeting the intended quality is the responsibility of all parties involved. Every party is responsible for the quality of their own subsystem and connecting interfaces as mentioned in section 3.2 above.

4.1 Product Risk Analysis

For every risk that is identified during the Product Risk Analysis, the chance of failure and the damage that would cause should be discussed. After that, a risk class (high, medium or low) is estimated per risk scenario, which will help to prioritize test cases.

The result of this PRA is presented in Table 4. This is the result of a preliminary inventory. The final PRA will be carried out with the consortium after the scope of each prototype is known.

Table 4: Product Risk Analysis

Risk	Failure chance	Damage	Risk class
Delay or incomplete deliverables of WPs; dependence of some WPs on data/deliverables from other WPs.	Low	High	Medium
Connections, interfaces and integration of all components.	High	High	High
System recovery in case of a disconnect of the single point of failure of the POWER2DM system: the Personal Data Store.	High	High	High
Corrupt data, database or archive data.	Medium	High	High
Incorrect versions installed or unwanted variation in configuration of different components.	High	Medium	High
Failure of components other than the PDS, does the system keep running?	Medium	High	High
KADIS and MARVEL performances, related to load.	Medium	Medium	High
Installation from scratch in case of hardware failure.	Low	High	Medium
PDS performance and stability.	Medium	High	High
Functionality and Usability of User Interface Applications (Shared Decision, Mobile and Web applications).	Low	Low	Low
Replacement of server or components in case of failure.	Low	Medium	Low
Lack of data management (cleaning and archiving). Failure dependent on amount of data.	Low	High	Medium
Lack of monitoring system for every component (?)	High	Medium	Medium

These identified risks can and should be taken into account when preparing the component specific and system chain test cases. For example: perform a failure and recovery test with the PDS. Depending on the risk identified, it can be translated into one or multiple test cases, checks (points of attention) or types of tests. The latter is also taken into account in the test strategy, addressed in section 4.2.

4.2 Test strategy

The test strategy should be based on risk management. The risks identified during the Product Risk Analysis can be used for this.

The following test types are defined:

Table 5: Test types

Abbreviation	Test Type	Description
UT	Unit Test	A test of a specific part of a component, usually done by the developer on his/her local system during development of the component and by nightly builds. This includes code review.
ST	System Test	A test that is done by a tester on a dedicated test environment. Test cases are designed from the specifications and requirements. This includes integration and interface testing.
I	Implicit test	A test that is executed implicitly without stepwise test cases, such as user friendliness.
A	Automated test	A test that is automated. Results of every automated test are sent to testers, developers and project management after the test has run.
R	(Document) Review	A review is a static test rather than a dynamic test, where focus lies on correctness and completeness.
FAT	Factory Acceptance Test	The FAT is done at premises and the tests themselves or results can be observed by the customer, where the functioning of the system serves as a go/no go for the next stage.
SAT	Site Acceptance Test	The SAT is done on a site (acceptance) environment, usually on site at the purchaser of the system. To check by means of inspection and dynamic tests, that the software can be installed properly, that connections and interfaces with external systems function and that the product works in its dedicated working environment.

In Table 6, for each test/quality the components, test type and strategy are filled in. The test strategy can be based on the results of the PRA. Test strategy can be a test technique, coverage type or other test method.

Table 6: Test strategy

Test / Quality			Subsystem(s)	Test Type	Test Strategy	Comment
☑	**Installation**					
	☐	**Installation manual**				N.a. Systems are monitored by developers.
	☑	**Configuration**	Configuration files	R	Check new configuration settings and that changes to configuration are applied correctly.	Differences between production and test environments should be as minimal as possible. Differences when changing settings should be checked with a text comparison tool.
	☑	**Installation**	All components	ST	Performing installation	

Test / Quality			Subsystem(s)	Test Type	Test Strategy	Comment
	☑	**Distribution package**	All software, configuration files and release notes necessary for installation.	R	Review	Check for correctness and completeness.
☑	**Regression**		All components	UT, ST, I, FAT	Functional process flows test cases, exploratory testing, error guessing, performance testing.	Every time a component and/or the complete platform is updated a regression test is carried out.
☑	**Connectivity**		All components	UT, ST, I, FAT	Check that data can be sent and received between components.	
☑	**Functionality**					
	☑	**Specifications**	Designs and storyboards	R	Review	
	☑	**Interfaces**	Between all components	ST, I, FAT	Check that data and content that are sent are completely received.	
	☑	**Logic**	All components and messages, logs and data necessary to check functionality.	UT, ST, FAT	Process flow testing, conditions and decisions coverage.	
	☑	**Correctness**	Messages, logs and data	ST, I, FAT	Check that sent, received, saved and or shown data during functional testing is correct and not lost or changed unless specified.	
	☑	**Completeness**	Messages, logs and data	ST, I, FAT	Check that sent, received, saved and or shown data during functional testing is complete and not lost or changed unless specified.	
☑	**Flexibility**			ST, I, FAT	Check that features that can be changed by	

	Test / Quality		Subsystem(s)	Test Type	Test Strategy	Comment
					users are functioning properly.	
☑	**Usability**					
	☑	**User interface**	Shared Decision, Mobile, Web Applications	ST, I, FAT	Syntactic test, semantic test and validation.	
	☑	**Help functions**	Shared Decision, Mobile, Web Applications	R	Review	
☑	**Continuity**					
	☑	**Robustness**	All components	ST, I, FAT	Error guessing. Stopping, restarting and killing components. Test with corrupt and invalid data. And inspection of logging.	
	☑	**Recovery**	All components	ST, I, FAT	Will be tested by performing several functional test scenarios after network connections, servers and/or components have been reconnected.	
	☐	**Failover**				N.a.
☑	**Performance**					
	☑	**Normal load**	All components	I, FAT	Simulating load based on realistic load profile on components.	Needed performance of system is not specified. Therefore, exploratory testing will be carried out. During functional testing limited load testing will be done implicitly.
	☐	**Stress**				Not deemed necessary
	☑	**Endurance**	All components	FAT	Simulating load based on realistic load profile on components, running for several days.	

	Test / Quality	Subsystem(s)	Test Type	Test Strategy	Comment
☐	**Infrastructure (suitability)**				N.a.
☐	**Portability**				Not deemed necessary
☐	**Reusability**				Not deemed necessary
☑	**Security**	Identity and Access Management Components	FAT	Penetration tests, will be tested by trying to access network, systems and (personal user) data that should not be possible to access without the required authorisation.	
☑	**End to end**	All components	FAT	Will be tested by performing several functional test scenarios from end to end.	

4.3 Test basis

The following documents form the high-level test basis for this project. The test cases should be based on these requirements and specifications. If any of the stakeholders wants to add or change specifications, this needs to be discussed and agreed upon and added to this documentation. More detailed documents, such as Prototype Storyboards, can be used to work out specific use case tests.

Table 7: Test basis

Ref #	Title	Author
1	POWER2DM D1.1 User Requirements and Use Case Scenarios	Wilma Otten (TNO) Hilde van Keulen (TNO) Pepijn van Empelen (TNO)
2	POWER2DM D1.2 Requirements Specification	Tuncay Namlı (SRDC)
3	POWER2DM D1.3 Conceptual Design	Tuncay Namlı (SRDC)

5 TEST APPROACH

5.1 Test approach

During the development of the software, system tests should be done whenever a developer delivers a test release of (part of) the software. The use of the TMap approach is highly recommended.

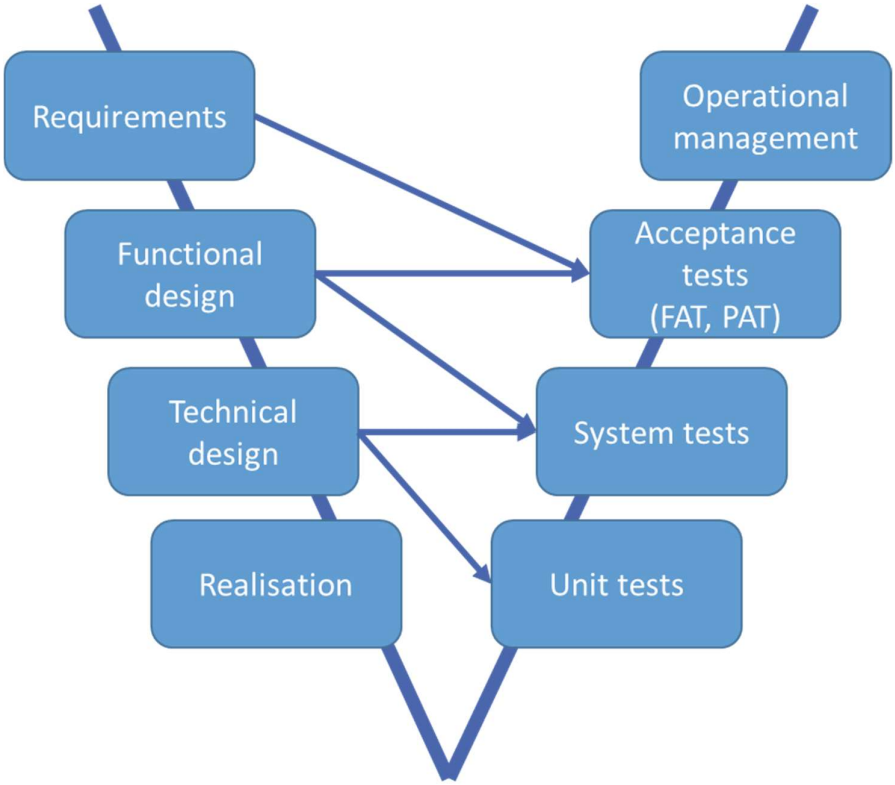

Figure 9, V-model

All parties should complete the V-model process from the Requirements phase up to System test phase before the delivery of their subsystems for acceptance testing.

Take into account the risks identified during the product risk analysis (section 4.1), the test strategy (section 0) and of course the test basis (section 0).

The system tests, including integration tests, will be concluded with a Factory Acceptance Test (FAT) before delivery of the system for conducting the pilot sessions.

5.2 Test specifications

In POWER2DM D1.1 and D1.2 the requirements for the software are described. These documents form the test basis for the test cases that will be executed to judge the quality of the software and whether the requirements are met.

D1.1 and D1.2 contain a set of personas and use cases that describe the functional system specifications in a detailed way. For the more technical subsystem tests and integration tests, additional test cases for the integration between the different Power2DM components and test data may have to be created by the responsible parties (see section 3.2).

During the FAT, use case tests are performed to test the integrated system “chain” of subsystems, based on the use cases provided in D1.1 and D1.2, taking into account the conceptual design choices made as described in D1.3.

All parties should deliver a test plan for their subsystem including the interface tests for connecting interfaces. A template for this will be provided by PD. This test plan should include at least the test strategy (test/quality attributes, test types, methods), as well as the specifications of the test environment(s) in which the tests will be performed. Any deviation from the general test approach described in this Master Test Plan should be clearly stated and clarified.

5.3 Issue registration and management

Issues should be registered in Redmine. PD will make a project for POWER2DM and can provide access to all required users.

For every issue, the build/version number of the component in which the issue has occurred must be registered. That is a mandatory field when registering an issue in Redmine. Besides this, a build version or due date on which the issue is expected to be solved should be filled in. When the issue actually is fixed and tested with a sufficient result, the build/version in which the issue is solved should be registered as well.

Defects should be classified in categories as described in Table 8 and should be given one of the corresponding priorities to make sure they will be fixed accordingly. Due to the complications a patient may encounter due to a defect in the system, the tester should consider having healthcare professionals (internists/GPs) prioritizing the defects.

Table 8: Defect categories and prioritization

Defect category	Severity	Defect prevents acceptance?	Priority in Redmine	How the defect is handled
Critical	The user/system is not able to perform the required function due to breakdown, unresponsiveness, loss or corruption of data. Or the function provides wrong results.	Yes	Immediate or Urgent	The defect will be worked on with high priority. Developer needs to stop what he/she is doing and fix and release immediately when ready.
Major	The user/system can operate the function, but is faced with unexpected difficulties such as unexpected displays, wrong or missing help or error messages, illogical or deviating workflow.	Yes	High	The defect will be worked on with regular priority, as soon as possible, a new release date is communicated to tester/customer.
Minor or Cosmetic	The user/system can operate the function facing minor difficulties such as click buttons deviating from specification or standards, spelling or text formatting errors, or other minor mistakes that do not influence the main functionality or flow of the system.	Open for discussion	Normal or Low	The defect will be worked on with low priority. If necessary, it will be fixed for a next planned release.

5.4 Reporting test results and requirements coverage

Together with the delivery of a subsystem, a test report must be delivered including the test cases performed, and the results of those tests. It should be shown that all requirements relevant for the developed component are covered. This can be done by showing the link between one or more requirements and test case(s). How exactly this can be done depends on the tools that are used by the different software development partners in the POWER2DM consortsium, but it can be done with most test management tools. PD uses SpiraTest, for example, in which requirements and test cases can be linked and the test results and requirement coverage can be shown.

If there are any open issues in a subsystem when it is released for end to end system testing, this should be clearly indicated in the test results report, made visible (for example by marking the related test case, requirement or comment with a colour), and a comment added with a short explanation of the deficiency and a reference to the registration of the issue in Redmine.

5.5 Acceptance criteria

The most important criteria for acceptance are that all test cases have been executed and a test report has been delivered. Not all test cases have to pass, it is possible for minor defects to exist that do not block acceptance. If the software is delivered with any open or known issues, this will be included in the test report. If, when and how these defects will be solved must then be discussed and agreed upon.

6 TEST PRODUCTS AND INFRASTRUCTURE DTAP

In this section, the environments and test products such as simulators and test data are specified.

Ideally, it's best to have a DTAP (Development, Test, Acceptance and Production) street, but it is expensive. Therefore, the following is proposed, for each software development party to have:

- A development environment to be able to develop the models and services
- A test environment to integrate and test the models and services
- A production environment for the pilots and evaluation

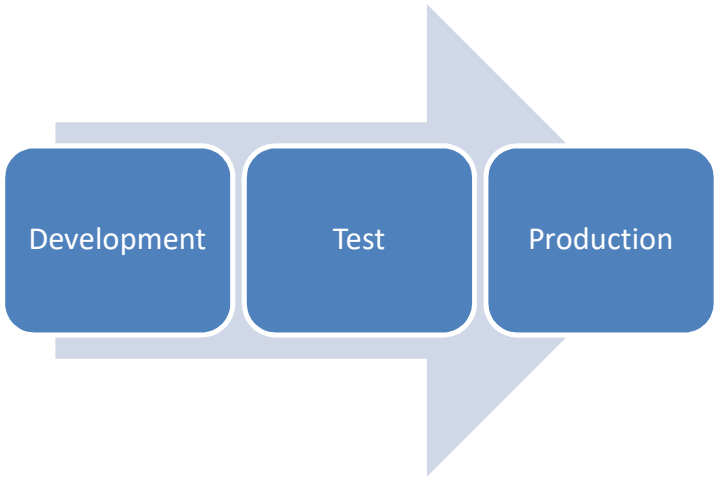

Figure 10, proposed DTP street

6.1 Management of D, T, (A) and P environments

Every party should host, administer, and manage their own environments.

The production system will be realized by connecting with each other's subsystems through the different web services from each parties' own production environment.

The same principle should be applied for the development and test environments.

6.2 Production environment

In this section the details about the production environment are specified based on the system requirements and hardware specifications to help to determine the software and hardware that is necessary for testing.

Each partner will host the component they are responsible for in their own local or cloud environment. PD will integrate these environments.

6.3 Development environment

Development occurs mainly on the local system of the developer. Developers will not run the entire system, but only the components that are necessary for the development of their own part, using stubs for the other subsystems.

6.4 Test environments

The main purpose of the test environment is for the integration of the subsystems of the different software development consortium partners into a complete system for chain testing and FAT. But besides this, each party should have their own (part of the) test environment for system tests of their own subsystem. Note that those test subsystems must be separated from the chain test system and may not influence the chain tests.

For example, PD will deploy the subsystems on different virtual machines on a test server in PD's data centre. This will be done in two ways, with different configurations. One for system testing, with stubs to simulate the other POWER2DM applications. And another for integration with the other parties' subsystems for chain testing.

6.5 Simulators and test data

Every party is expected to develop their own stubs and/or simulators for the other subsystems, and test data, as previously mentioned in section 3.2. Simulators, stubs and/or test data may be shared between the software development partners of POWER2DM if this is helpful.

7 ORGANISATION

In the table below the resources and roles of the members of the project team are presented.

Table 9: Resources and roles

Role	Name(s)
Project leader	Marlies Schijf (PD)
Lead developer	Teams at SRDC, PD, iHealth, SRFG, TNO, IDK
Developing partners	SRDC, PD, iHealth, SRFG, TNO, IDK
Test manager	Marlies Schijf (PD)
Lead tester	Roosmarijn Schopman (PD)
Test team	Teams at SRDC, PD, iHealth, SRFG, TNO, IDK

7.1 Version control

Each party is responsible for the version control of the software for the subsystems they develop.

7.2 Release management

Each party is responsible for the deployment of their own software (for the subsystems they develop) in the test environment as well as the production environment. However, the timing that software may be updated has to be agreed upon between each party and the project test manager (as indicated in the table above).

For every release, a deployment package is created by the party responsible for the subsystem software. This package contains the latest components with unique version numbers.

For every release, the responsible party should send a release mail with the software components that are ready for release, version number per component and release notes describing the changes and issues fixed if relevant.

For deployment in the test environment, each party should discuss and agree with the test manager when and what may be deployed.

For deployment in the production environment, the software first of all has to be accepted by the lead tester, and when the software may be deployed should be agreed upon between all partners and the project managers.

8 PLANNING

Below are the release cycles for the POWER2DM system. Two releases are planned in 2017 and two are planned in 2018. Further below is a figure with the detailed planning of the development of prototype I. Finally, this chapter contains a table with the activities planning.

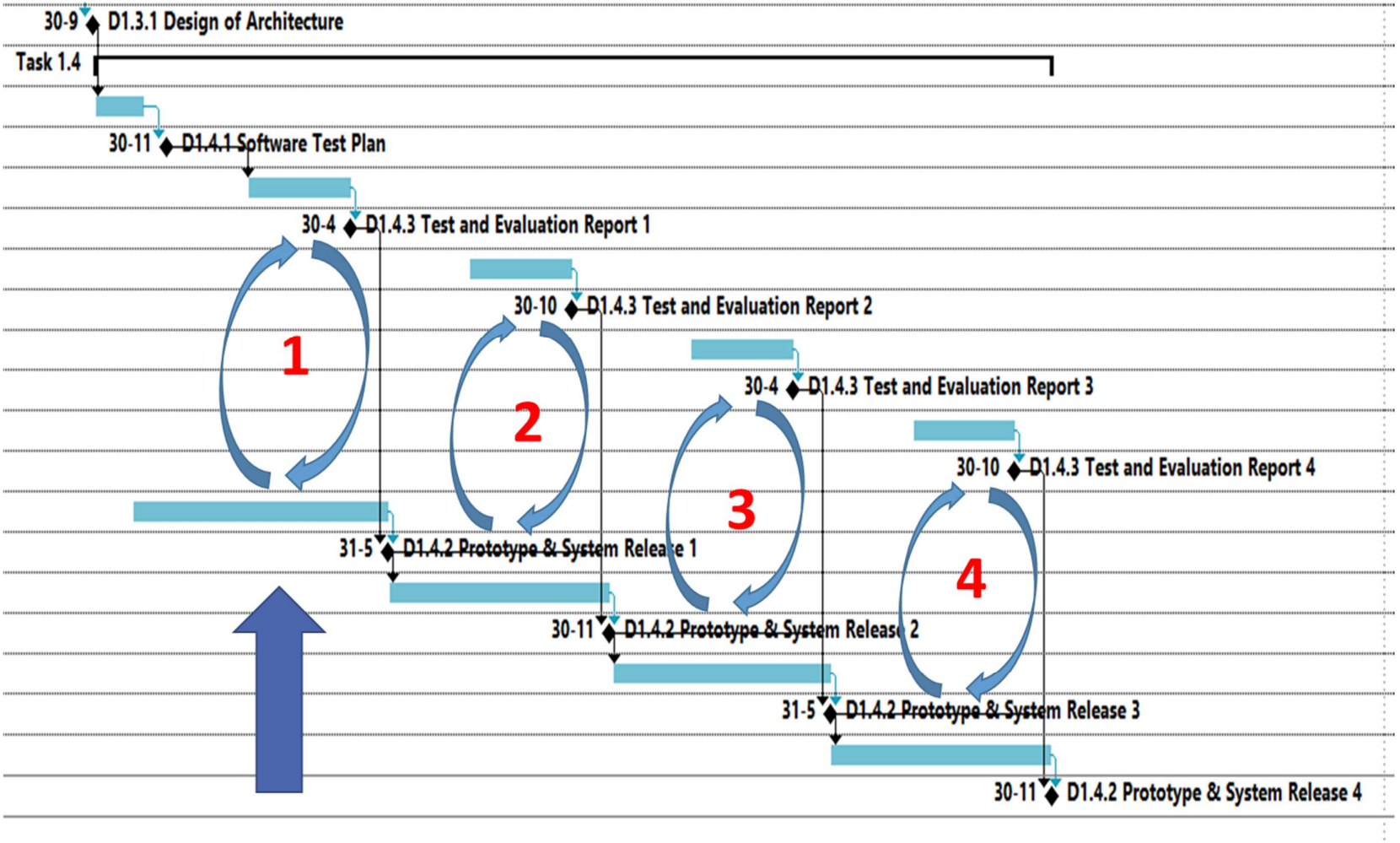

Figure 11, Planning of release cycles

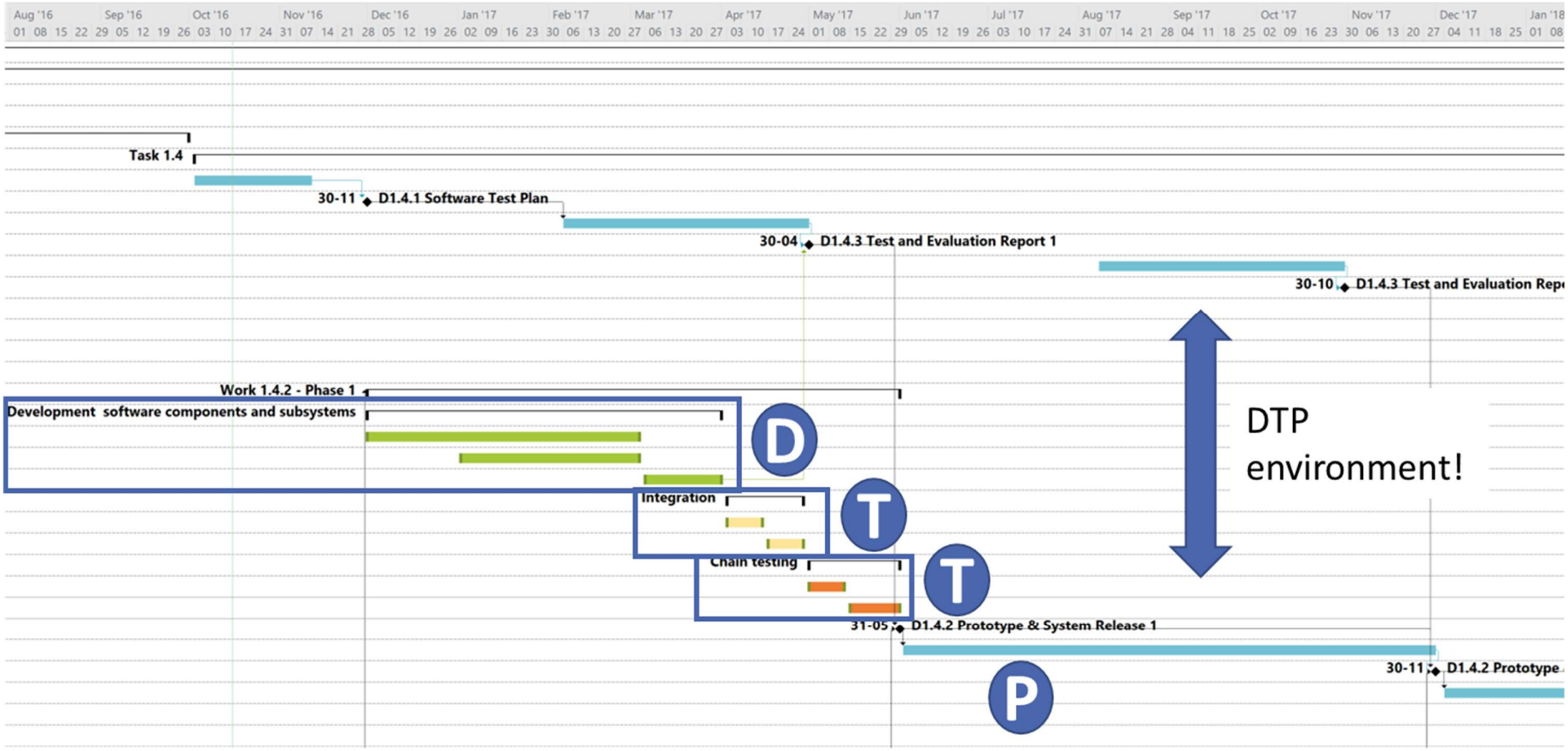

Figure 12, Detailed planning of first prototype development

Table 10: Activities planning

Task	Who	Start Date	End Date
Product Risk Analysis	All stakeholders	09-01-2017	15-02-2017
Update Test Strategy	Lead test engineer	09-01-2017	15-02-2017
Development of subsystems	Developers SRDC, PD, iHealth, SRFG, TNO, IDK	Currently	
Development of stubs and simulators for testing	Developers SRDC, PD, iHealth, SRFG, TNO, IDK		15-01-2017
Redmine project page created	Test manager		15-01-2017
Test plan per subsystem ready	SRDC, PD, iHealth, SRFG, TNO, IDK		15-01-2017
Test environments ready	SRDC, PD, iHealth, SRFG, TNO, IDK		15-01-2017
Test tooling ready	SRDC, PD, iHealth, SRFG, TNO, IDK		15-01-2017
Test data ready	SRDC, PD, iHealth, SRFG, TNO, IDK		15-01-2017
Releases of subsystems for testing	Lead releasers SRDC, PD, iHealth, SRFG, TNO, IDK		15-01-2017
System testing of subsystems	Testers SRDC, PD, iHealth, SRFG, TNO, IDK	16-01-2017	29-01-2017
Bug fixing	Developers SRDC, PD, iHealth, SRFG, TNO, IDK	30-01-2017	05-02-2017
Retesting of subsystems	Testers SRDC, PD, iHealth, SRFG, TNO, IDK	06-02-2017	
Releases of subsystems to test for integration	Lead releasers SRDC, PD, iHealth, SRFG, TNO, IDK		13-03-2017
Test results reports per party	Testers SRDC, PD, iHealth, SRFG, TNO, IDK		12-02-2017
Integration of subsystems	Developers SRDC, PD, iHealth, SRFG, TNO, IDK	13-03-2017	
System Integration testing	Testers SRDC, PD, iHealth, SRFG, TNO, IDK	27-02-2017	
Bug fixing	Developers SRDC, PD, iHealth, SRFG, TNO, IDK	06-03-2017	
System Integration done	SRDC, PD, iHealth, SRFG, TNO, IDK		12-03-2017
FAT	Lead tester PD	13-03-2017	
FAT results report; software OK?	Lead tester PD		19-03-2017
Bug fixing	Developers SRDC, PD, iHealth, SRFG, TNO, IDK	20-03-2017	02-04-2017
Subsystems regression testing	Testers SRDC, PD, iHealth, SRFG, TNO, IDK	03-04-2017	
Regression test results reports per party	Testers SRDC, PD, iHealth, SRFG, TNO, IDK		16-04-2016
End to end regression testing	Lead tester PD	17-04-2017	
Test and Evaluation report 1	Lead tester PD		30-04-2017

System release 1	SRDC, PD, iHealth, SRFG, TNO, IDK		31-05-2017
…			
Test and Evaluation report 2	Lead tester PD		30-10-2017
System release 2	SRDC, PD, iHealth, SRFG, TNO, IDK		31-11-2017
…			
Test and Evaluation report 3	Lead tester PD		30-04-2018
System release 3	SRDC, PD, iHealth, SRFG, TNO, IDK		31-05-2018
…			
Test and Evaluation report 4	Lead tester PD		30-10-2018
System release 4	SRDC, PD, iHealth, SRFG, TNO, IDK		31-11-2018