Co-funded by the Horizon 2020
Framework Programme of the European Union

HORIZON2020 FRAMEWORK PROGRAMME

ICT – 21 -2014

Advanced digital gaming/gamification technologies

ProsocialLearn

Gamification of Prosocial Learning

for Increased Youth Inclusion and Academic Achievement

D2.3

1st System Requirements and Architecture

Document Control Page

WP/Task	WP2 / T2.1
Title	D2.3System Requirements and Architecture
Due date	01/07/2015
Submission date	05/08/2015
Abstract	This document present the first version of the ProsocialLearn architecture covering the principle definition, the requirement collection, the “business”, “information system”, “technology” architecture as defined in the TOGAF methodology.
Author(s)	S. Modafferi (IT Innovation); S. Crowle (IT Innovation)
Contributor(s)	L. Middleton (IT Innovation); K. Dimitropoulos (CERTH); E. Dimaraki (EA); S. Cobello (EUR); F. D’Andria (ATOS); J.M. Garrido Gonzalez (ATOS); M. Boniface (ITINNOVATION)
Reviewer(s)	K. Dimitropoulos (CERTH), Francesco D’Andria (ATOS)
Dissemination level	☐ internal ☒ public ☐ confidential

Document Control Page

Version	Date	Modified by	Comments
0.1	25/06/2015	Stefano Modafferi	First draft
0.2	24/07/2015	Stefano Modafferi, Simon Crowle, other partners	Revision according to the feedback and discussion of the GA held in Thessaloniki on 29/30 jun 2015
0.3	31/07/2015	Stefano Modafferi	Final version. Implementation of comments received on the 0.2 version
1.0	03/08/2015	Stefano Modafferi	QAd version implementing requested change from internal reviewers

List of Abbreviations

Abbreviation	Description
API	Application programming interface
ASM	Active Shape Model
AU	Action Units
CDP	Cloud Design Principles
DCRM	Datacentre resource management
DGM	Deployable Game Model
FACS	Facial Action Coding System
GUI	Graphical User Interface
IaaS	Infrastructure As A Service
LAN	Local Area Network
LMS	Learning Management System
LUT	Look Up Table
MVC	Model View Control
NPC	Non Player Character
PaaS	Platform As A Service
PLO	Prosocial learning objective
PSDP	Prosocial Design Principle
PSL	ProsocialLearn
QoE	Quality of Experience
QoS	Quality of Service
RDF	Resource Description Framework
SaaS	Software As A Service
SSO	Single Sign On
TOGAF	The Open Group Architecture Forum
Core domains	Different aspect characterizing prosociality (cf. D2.1)
Prosocial measurable context	The context within which a Prosocial measurable space is defined.

Prosocial measurable space	A metric (measurable) space related to a core domain where a state can be assessed and an objective set.

Executive summary

This document presents the first version of the ProSocialLearn architecture, also listing the main requirements so far collected. It provided the set of models and component developed for operationalising the psychological framework described in D2.1. An ontology (see Figure 9 - ProSociaLearn Ontology) summarizes all the relationships among concepts. Use case diagrams, class diagrams, component diagrams and data flow diagrams highlight all the different aspects of the architecture.

Index

1 Introduction

1.1 Purpose of the document

This document presents the first version of the ProsocialLearn Architecture. The description follows the TOGAF® methodology[1].

The document is the first attempt to create a global coherent view across the research activities in the different work packages. In particular the operationalization of the psychological models presented in D2.1 is one of the most valuable outputs of this document

The principles upon which the architecture is built are highlighted and the technical requirements described. The requirements come from different sources and a discussion for all of them is presented.

The architecture is broken down according to the TOGAF methodology in architecture vision, “business”, “information system” and “technology” architectures. Some of the model and modules described are still under definition and therefore the maturity of the different parts is not aligned.

Because of the relatively early phase of the project (M6), within the general goal of designing the system’s architecture, the focus in this document has been on the requirements and on the data modelling. The next iteration of the document (due at M15) will complement and revise the current version providing all the necessary clarification.

1.2 Scope and Audience of the document

The document is intended for general audience willing to get an overview of the complex systems of models and component characterizing the ProsocialLearn scenario.

It is also intended for software developers that will implement the different components, to understand the relationship and interfaces of these components with the others.

1.3 Structure of the document

This document follows the TOGAF methodology where “business”, “information system” and “technology” views on the architecture are separately presented to increase clarity of the foreseen solution. In particular Section 2 Architecture Design Principles shows the design principle inspiring the ProsocialLearn architecture, Section 3.1 presents an high level view introducing the main ideas and concept, Section 3.2 deals with requirements, Section 3.3, 3.4, and 3.5, respectively present the “Business”, “Information System”, and “Technology” architectures while in Section 4 - Conclusions we summarize the conclusion.

[1] https://www.opengroup.org/togaf/

2 Architecture Design Principles

According to the TOGAF methodology the definition of design principles is addressed in the Preliminary phase. The architectural principles driving the design of the ProsocialLearn (PSL) architecture are based on the two complementary aspects present in the project: the ability of hosting 3rd party games and the ability of providing prosocial features to them. The ability of hosting games supporting their scalability is particularly associated with the cloud principles defined by NIST[2] and Hewlett Packard[3] (identified as CDP) while specific principles cover the offer for prosocial features (identified as PSDP, ProSocial Design Principles).

	Principles	Description	ProsocialLearn context
CDP1	On-demand self-service2	"A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider".	This principle states that a game provider shall be able to automatically deploy the game on the PSL platform in a PaaS fashion.
CDP2	Broad network access2	"Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations)".	The PSL platform is accessible in a secure way through the standard internet and well defined APIs.
CDP3	Resource pooling2	"The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand"	PSL supports the efficient deployment of cloud governance of a gaming application on a cloud infrastructure leveraging cloud APIs and added value functionalities.
CDP4	Rapid elasticity2	"Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with	PSL platform will automatically scale up and down (horizontally and vertically) cloud resources to maximize the performance of each module of the gaming application; it restarts and replaces failed components.

[2] National Institute of Science and Technology, The NIST Definition of Cloud Computing. Retrieved 30 March 2015 http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[3] HP, Seven design principles for building a converged cloud. HP white paper 26 July 2013. Retrieved 30 March 2015. http://www.hr-insider.com/content28976

	Principles	Description	ProsocialLearn context
		demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time".	
PSDP1	Requirement capturing	A robust process of requirement capturing is crucial for defining the prosocial features offered by the platform	The project consortium includes several stakeholders (e.g. game developers and schools' representatives) that are an active part of the process of requirement collection.
PSDP2	User Centric and different stakeholder viewpoints	Well-designed services and user interfaces are key to deliver the best user experience.	PSL deals with many different types of users: game providers to support their game deployment, teachers, carers and possibly scientists through the dashboard and in some situations also the students. For each of them PSL designs suitable interfaces considering the different kind of interaction they have with the platform.
PSDP3	Security/ privacy (data protection)	Security and Privacy should be always ensured by the platform.	PSL implements privacy by design approach, so has been considering it since the early phase of the design.
PSDP4	Ethics	The Information systems and the security of information systems should be provided and used in such a manner that the rights and legitimate interest of others are respected	PSL implements a rigorous process for covering ethical aspects and to the platform reflects requirements coming from the ethics and in general legal aspects of having games where kids are involved.
PSDP5	Compatibility and game independency	The API should be general enough to ensure the maximum compatibility with game engines	The ProSocial API is defined to support any kind of game and also the underlining prosocial model is general enough to enable its use by many different games.

Table 1 - ProsocialLearn design principles

3 Architecture description

PSL uses the TOGAF® methodology[1] to describe the architecture. An initial section presents a high level view of the main characteristics (corresponding to the Architecture vision in TOGAF). Then the requirements derived from other activities in the project are presented and finally the architecture is described according to three different views: Business Architecture; Information System Architecture; Technology architecture.

3.1 Architecture high-level view

Figure 1 - Simplified architecture shows a simplified view of the ProsocialLearn architecture.

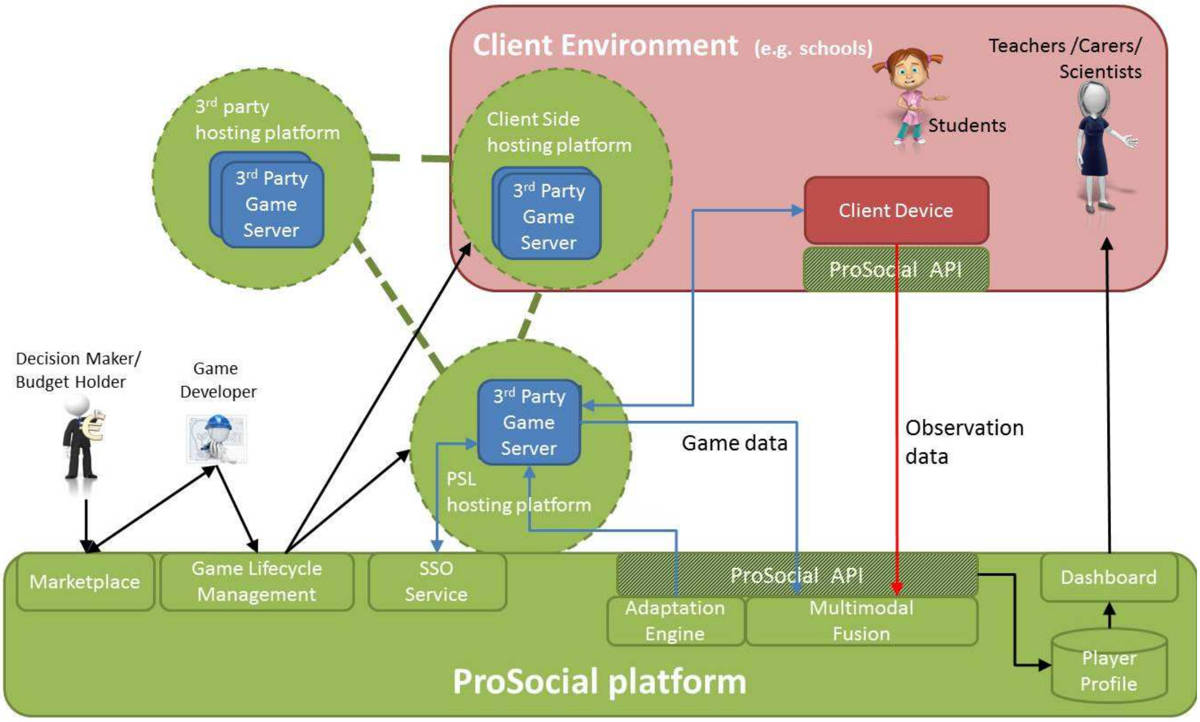

Figure 1 - Simplified architecture

It is composed of three main parts:

- The Client Environment. This part of the platform represents the environment where the clients are deployed (e.g. the LAN of computers in a school). It is assumed that PSL can install software in the Client Environment to support communication with the game server and with the platform for the acquisition and analysis of observations.
- The Hosting Platform. This part of the ProsocialLearn Platform is responsible for hosting third party games. It can be described as a cloud-like platform offering scalable provisioning of game hosting services. In reality, as shown in Figure 1 - Simplified architecture the game server can be deployed in the platform server side, in the client environment or even in a third party cloud facility. This possibility is enabled by the service paradigm implemented by the platform.

- The ProSocial Platform. This is the core of the platform. It has the following main features:
  - performing the analysis to assess the prosocial state of a player through multi-modal fusion of observations and game state;
  - suggesting the adaptations necessary to optimize prosocial learning and
  - providing an interface for teachers, scientists and carers to browse related student profiles.
  - Providing a marketplace for budget holders and game developers.

The ProSocial platform also offers a Single Sign On Service (SSO) to the game engine so that players can be profiled across multiple games. Sensitive information is filtered, simply providing to the game servers the necessary and privacy compliant views.

A set of ProSocial APIs that allow a game engine to interact with the platform is also defined. It is important to highlight that the ProsocialLearn approach is not limited to information related to the game status. In fact the assumption to be validated by scientific evidence during the project lifetime is that by observing players' behaviour it is possible to collect different and useful information to be fused to infer a prosocial state for the player and that it is possible and useful to leverage on this information to decide how to adapt the game with the final goal of teaching students to be more prosocial.

Observations include therefore in game data and externally (with respect to the game) observed information, which is any information that is possible to collect outside the game that is meaningful to infer players' current states. In particular being able to identify players' emotion and engagement is believed to have a role in choosing the best game adaption for the specific situation.

Processing external observations requires careful consideration because of ethical and legal constraints that may exist in environments like schools. PSL facilitates the handling of these issues supporting both the local analysis of the data on the client side and on the server side.   shows an example of observations.

Observation type	Description
Voice	time, various sound features (energy in frequency band, pitch, duration)
Mouse/touch motions	location and click speeds, time, x, y
Facial expression	Time, facial features, facial action units
Chat	Items from chats with a common understood meaning

Table 2 - Example of Observations

Figure 2, shows the relationship among observations, in game data and player profile information. They all are processed in order to extract relevant information and then algorithms specifically defined across the prosocial core dimensions defined in D2.1 are used to infer the current prosocial state defined on a prosocial model. At any moment in time for a given core domain, a player will be in a specific point (assessment phase) and goals (ProSocialLearn objectives) can be set for him a priori or at run-time. The game will then adapt to try to lead the player to move his current position in the prosocial space towards a more desirable one.

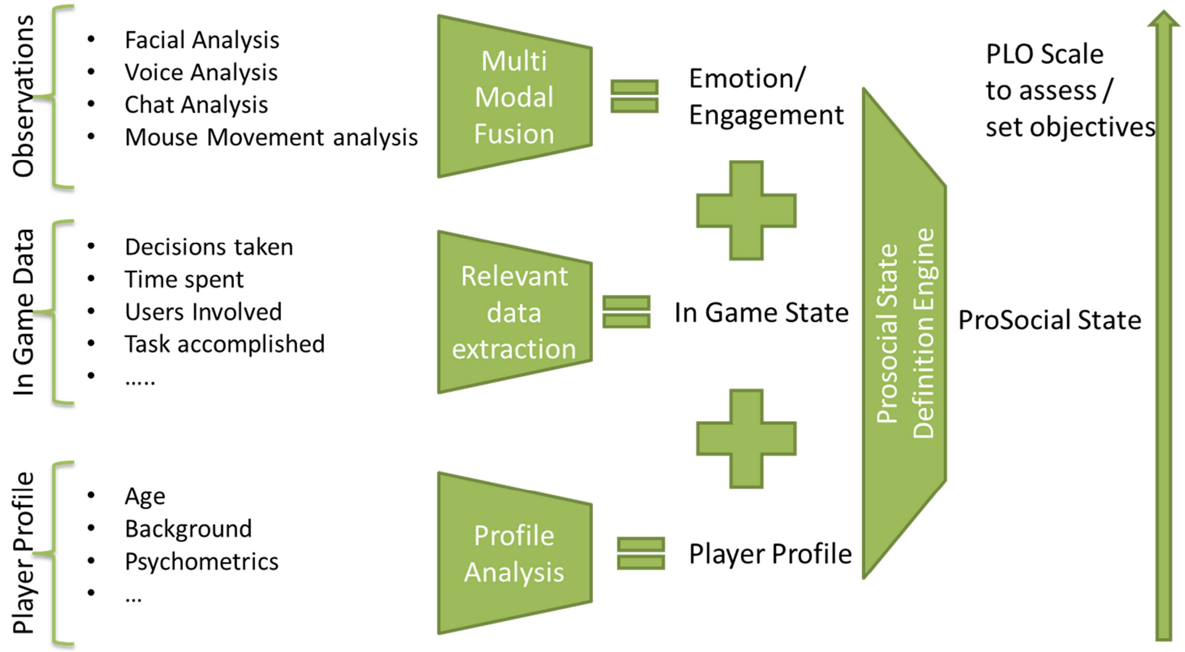

Figure 2 - Relationship among observations, in game data and player profile

3.2 User requirements

User requirements in PSL come from different sources. The market analysis performed in WP1 and reported in D1.1[4] sets the scenario where the platform will operate. The user requirement analysis for prosocial learning reported in D2.1 describes what the prosocial learning objectives are and how they may be related to on-line games. The requirements related to ethical aspects are defined in D7.1[5] and an independent board is in charge of overseeing the experiments, so also impacting the architecture. The teachers' and careers' perspective is captured through the interaction with school representatives in the consortium and taken into account in implementing the dashboard component.

ProsocialLearn performs requirements management using a simple common template (e.g. see Table 3 - Technical requirements derived from the market business analysis) and a shared document. Periodic validation of new or changed requirements is performed in a dedicated technical telco as per the chosen agile approach. Priorities are assigned after discussing them with the experts present in the consortium.

The following paragraphs analyse the impact of the requirements on the architecture identifying a set of technological requirements to be fulfilled by the architecture and its components.

[4] ProsocialLearn Consortium. D1.1 Market and Competition Analysis. 2015.

[5] ProsocialLearn Consortium. D7.1 ProsocialLearn ethical oversight procedures. 2015

3.2.1 Impact from Market and competition analysis

D1.1 identifies barriers and drivers for the adoption of ProsocialLearn games. Among them some impact the architecture. In particular on a technological topic:

- *"For social games, middleware technologies and platforms are important resource (driver) because they allow developers without in house technologies to produce games that match leisure game quality. Similar middleware technologies on the ProsocialLearn platform could enable developers to produce high-quality games to develop prosocial skills."*
- *There is "Scope for a specialist platform to cut production costs. Since collaboration between game designers and professionals in application domains is in its early days, the tools and production process is far from streamlined. Special purpose tools that facilitate game creation for the prosocial skills sector, embedding both game design expertise, management of media assets, evaluation tools, and pedagogical elements can improve the speed and quality of production, and reduce costs."*

While in the Supply-side capability:

- *"For social games, middleware technologies and platforms are important resources (driver). They allow developers without in house technologies to produce games that match leisure game quality. Similar middleware technologies on the ProsocialLearn platform could enable developers to produce high-quality games to develop prosocial skills."*
- *Opportunity to capitalise on downstream restructuring (driver). There are signs that issues related to hosting platforms, distribution, marketing and deployment of digital serious games are being tackled with the aim to structuring and 'pooling', at best in a standardized framework, downstream in the value chain. […] digital serious games is a cross-platform industry.*
- *Opportunity to bring scientific rigour to gaming metrics (driver). While currently products are in the main deployed on personal computers, it will certainly expand onto new generation consoles, and mobile and online platforms. Metrics used to optimize online gaming and maximize revenue can be used instead to evaluate use and behaviour and maximize impact. However this needs to be done in a much more scientific manner with goals of learning, behaviour change etc that go well beyond customer loyalty or repeat spending, and with considerable care over interpretation."*

These drivers influence the design of the PSL architecture. In particular they are translated in the following set of technological requirements where the "m" in the label identifies the class (m stands for market):

#ID	Description	Priority[6] [Must | Should | Could]
m.REQ1	The platform hosting the game should not be limited to support pure cloud style deployment, whereas the services offered should be more in line with current game platform (e.g. Steam7) where an ecosystem of different services allows the game developers and the end users to	Must

[6] https://en.wikipedia.org/wiki/MoSCoW_method

[7] http://store.steampowered.com/

	easily find what suits them.	
m.REQ2	With reference to what is described in REQ1, specifically a service of Single Sign On (SSO) facility must be available.	Must
m.REQ3	The interaction between game and the Prosocial Platform must be based on a shared vocabulary explaining the prosocial concepts and of a set of generic APIs matching game mechanics |with prosocial dimensions.	Must
m.REQ4	The offered solution must be scalable to support increasing game demand in the most possible transparent way for the game developers.	Must
m.REQ5	The marketplace where end-user (e.g. teachers or school or budget holders) can browse and find games should be designed to be as clear as possible and easy to use.	Should
m.REQ6	Role definition in the platform must exist for supporting different business models.	Must

Table 3 - Technical requirements derived from the market business analysis

3.2.2 Impact from User requirement analysis for prosocial learning

Analysis of prosocial dimensions and requirements from a psychological point of view are addressed in D2.1. An engineering process is necessary to apply what was described there to a prosocial platform that needs to provide automatic mechanisms for assessing prosocial states and setting prosocial goals. This process, i.e. in other worlds the design of the prosocial API, is one of the key goals for the project and its definition is covered in many work packages, mainly WP2, WP3 and WP4. The requirements definition reported in this section derives from a snapshot of the research as currently identified and everything requires validation from data evidence and further refinement.

A requirement for the platform is therefore to make available a significant quantity of data for the scientists (mainly psychologists) to validate the approach and, above all, its operationalization.

From a more technical point of view the requirements concerning the richness and accuracy of the models measuring the prosocial space with specific attention to the intrinsically fuzzy nature of that space and the rigour in specifying the boundaries within which the prosocial state assessment and objectives have been set. A clear understanding by the game designer of the prosocial mechanism is required and instruments to support it are necessary. Finally it is worth noting that some prosocial dimensions predicate on multiple occurrences of phenomena (games) and therefore a proper behaviour history should be created and made accessible.

In particular is the following are needed (the “p” identifies the class of requirements as prosocial):

#ID	Description	Priority [Must | Should | Could]
p.REQ1	Setup of a clear vocabulary explaining the type of prosocial mechanisms and how they are measured.	Must
p.REQ2	The prosocial properties (e.g. fairness attitude) in the player profile should be created following the same methods and metrics used in the games.	Should

p.REQ3	The possibility for a game designer to self-assess what a game is addressing in terms of prosocial core domains	Should
p.REQ4	Definition of stateful properties (e.g in the player profiles) related to mechanisms for measuring prosocial states (e.g. the average time spent on a game or class of games by a player over the single instance and over multiple runs)	Must

Table 4 - Requirement derived from the User requirement analysis for prosocial learning

3.2.3 Impact from Ethics

A formal procedure for being ethically compliant has been defined in D7.1. From a technical point of view the impact of the ethics on the architecture deals with data management and process. A secure platform is the base and has been envisaged since the proposal stage. Privacy should also been enabled in the platform. In particular it is necessary to

- support data anonymization, so that the association between the identity records and other information cannot be inferred if data were to get stolen or accidentally disclosed in some way outside the platform;
- provide opt out options that allow specific observation acquisition features (like video or audio) to be disabled on request;
- distinguish between the data accessed by the games hosted on the platform and by the prosocial platform itself that may have access to more accurate and sensitive information upon which it can create a simplified view to be provided to the games; and
- enable user profile browsing only to authorized person like teachers and carers.

The compliance with the data management requirements suggests that it may be easier if processing tasks are moved to the LAN in the Client environment. The architecture should therefore support this option ("e" stands for ethical).

#ID	Description	Priority [Must | Should | Could]
e.REQ1	All the features provided by the platform related to the acquisition of observations must be configurable and it should be possible to disable them on a player-by-player basis.	Must
e.REQ2	A game view on the player profile must be defined such that more sensitive information is reserved.	Must
e.REQ3	Provide role-based access to the player profiles. Instances of the roles must be related to the player identity (i.e. only a teacher of a given student can access his or her data; that is having the role of teacher does not enable browsing across all available student data.)	Must
e.REQ4	Data for student identity and data modelling their interaction between the games and the platform must be decoupled, so that they can be related only on the platform, while the association is not possible outside of it.	Must
e.REQ5	The design of the architecture should enable where possible the execution on the client environment of code in charge of observation analysis, so simplifying the management of sensitive data	Should

e.REQ6	End users will be able to set their privacy settings, by indicating whether their personal information to be used for improving the system and for research	Must
e.REQ7	End users should be able to review the personal information that the system collects	Should
e.REQ8	End users should be able to modify personal information, e.g. teachers should be able to erase student profiles, when the use of the system is completed	Should

Table 5 - Requirement from Ethic Management

3.2.4 Impact from the platform end-users

The complex environment where the PSL platform is going to operate is includes many actors with different roles and different requirements. The requirements derived by the market analysis (cf. Section 3.2.1) partially overlap with this section where we therefore focus on specific requirements not already defined above.

- Game developers require well-defined and easy to use interfaces and platform performance to be sufficient for not interfering with game experience.
- Teachers want to have clear and possibly easily configurable games; they also require easy interfaces where the prosocial status of students can be assessed and prosocial objectives can be set. Finally they want the possibility of browsing their student profiles.
- Scientists and psychologists need access to aggregated anonymous data for validating the scientific hypothesis underlying ProsocialLearn project.
- Budget holders and decision makers need a clearly designed marketplace where the offer of the platform is highlighted and an easy and flexible way of purchasing games is supported.

There is not a formal deliverable for collecting these requirements and therefore Table 6 - Requirements from the end-users shows all of them.

The above considerations can be translated into the following requirements:

#ID	Description	Priority[8] [Must | Should | Could]
u.REQ1	Interfaces for game deployment should be as standard and easy to use as possible	Must
u.REQ2	ProSocial API should be clear and well documented, stable and the interaction protocol should be as standard as possible.	Must
u.REQ3	Lightweight game configurability mechanisms should be made available for teachers	Should
u.REQ4	Teachers and carers should be able to browse related students profiles.	Must
u.REQ5	The dashboard should be clearly and well designed and the assessment and definition of respectively prosocial state and objective should be easy to identify.	Must

[8] https://en.wikipedia.org/wiki/MoSCoW_method

u.REQ6	It should be possible to define objectives as relative increment/decrement of the current assessment	Could
u.REQ7	Scientists and Psychologists need to be able to validate the scientific hypothesis underlying the project by having evidence on how the data are collected and having the possibility of browsing the collected data in an anonymous way.	Must
u.REQ8	The marketplace should take care of supporting multiple business models so as to suit the biggest possible number of budget holders.	Should
u.REQ9	Student profile: visual representation of progress on various dimensions that make it easy to consult quickly. Additionally, it should use colour to highlight areas that require teacher attention	Should
u.REQ10	Student profile: clear explanations on the meaning of student profile items provided upon teacher demand (e.g. by clicking on an item). These explanations should include recommendations for games	Must
u.REQ11	Ease of Use: fast and intuitive setup of games. Ideally games should be playable right away when assigned without configuration or consulting documentation	Must
u.REQ12	Info for teachers: games should be accompanied by a clear summary as well as clear information about game play, duration, number of players, Prosocial Learning Objectives (PLOs) other learning objectives. A minimum of such information should be required for any game introduced in the marketplace, to facilitate teachers' selection of appropriate games	Must
u.REQ13	Info for teachers: games should be accompanied by a teacher guide that include suggested lesson plans	Should
u.REQ14	Info for teachers: A general teacher guide on prosociality, prosocial concepts and PLOs should be available for teacher use and referenced through the games (e.g. being able to find the definition of 'empathy')	Must
u.REQ15	Ease of use: Fast and intuitive setup of user profiles including batch registration	Must
u.REQ16	Ease of use: Grouping of student profiles in lists for easy reference	Must
u.REQ17	Ease of use/Learnability: For standard functions (e.g. managing students, assigning games etc) the teacher interface should be similar to commonly used LMS	Should
u.REQ18	Context of use: Fast loading times over standard internet connections: no major waiting either for the platform or the games	Must
u.REQ19	Context of Use: Within a school a student profile persists across different teachers	Should
u.REQ20	Games that can be completed within 5-25 minutes, or that can reach a meaningful wrap point in 5-25 min	Must
u.REQ21	Context of Use: Games playable from home and on student's own devices	Could
u.REQ22	Personalize: Teachers have access to a history of games used and to a personal Library of games they use often	Could

u.REQ23	Teacher input/Community: teachers can rate games and tag games with information about how they used them: e.g. in what class, for what subject, a small description of their lesson. This information is viewable by other teachers	Must
u.REQ24	Teacher input/ Marketing: teachers who do not themselves make purchasing decisions will have an interface for influencing purchasing decisions, like a wish list	Should
u.REQ25	Marketing: In the teacher view, the system will provide game recommendations and highlights based on: most commonly used, highest rated, newly added, relevance to other games selected, divergence from other games selected [serendipity principle]	Should
u.REQ26	Search: Available games will be searchable by specific parameters and by word-search	Must
u.REQ27	Student Input: Students will be able to rate their enjoyment of games and that information will be in their profiles and available. This information will also be available in the platform in an aggregate form	Should
u.REQ28	Student Input: Students will be able to rate games in terms of their relation to various prosociality concepts and that information will be in their profiles as useful information for the teacher	Could
u.REQ29	Student access to information: Students will be able to see an appropriate view of their personal profile.	Must
u.REQ30	It is necessary to define clearly all the necessary equipment to play in order to set up a database of available tools on the market	Must
u.REQ31	It is important to organize the pedagogical guidelines for teachers and underline the main aspects of system improvements in a defined period	Should

Table 6 - Requirements from the end-users

3.2.5 Other technical requirements

Other technical requirements may arise in defining the interaction among modules in the platform. These requirements are more fine-grained and technically described. They are not reported in this document, but they are part of the live (i.e. continuously evolving) document used for managing the requirement collection and validation.

3.3 Business architecture

Figure 3 - Value Chain in the ProsocialLearn scenario shows the value chain in the PSL scenario as already identified in the Description of work. Motivation, background and analyses (e.g. SWOT[9]) are presented in D1.1. It is reprised here for highlighting the complexity and multifaceted environment where the value is not limited to the money, but knowledge and teaching prosocial behaviours is key for some of the identified stakeholder. In fact value is expressed in terms of economic and societal transactions. Customers (schools or funding bodies) subscribe to games produced by Game Developers that are delivered by the ProsocialLearn Platform. The Platform builds on Gamification

[9] https://en.wikipedia.org/wiki/SWOT_analysis

Technologies developed by Technology Providers. A proportion of revenue flows from the Customer to Technology Providers and Game Content Partners. ProsocialLearn will deliver games to schools using a SaaS/Paas model. This provides the means to access European scale markets and lowers the barriers to adoption for schools by moving capital infrastructure, IT support and upgrade costs to the Platform provider.

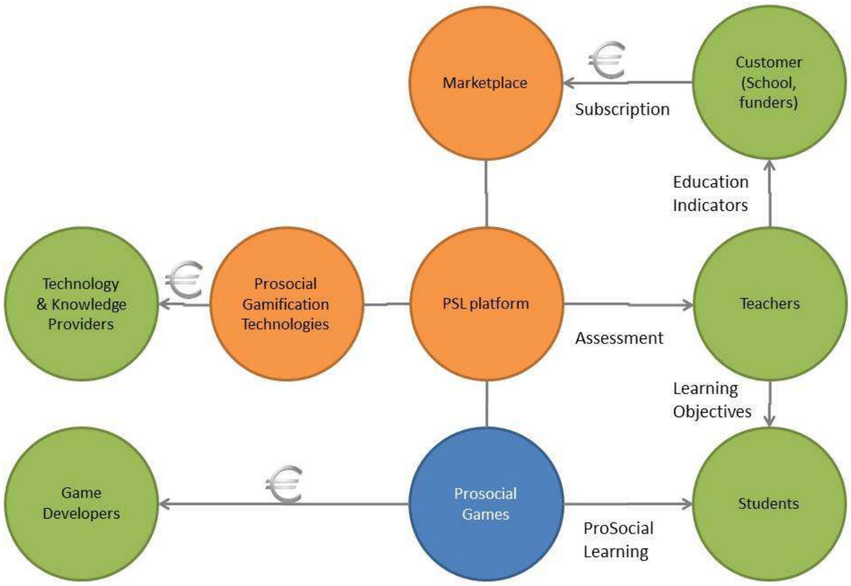

Figure 3 - Value Chain in the ProsocialLearn scenario

3.3.1 Use Case diagram

Leveraging on the analysis of the value chain above and on the requirement collection activity, a level 0 Use case model is presented in Figure 4. It is possible to identify all the users and the main use cases of the platform.

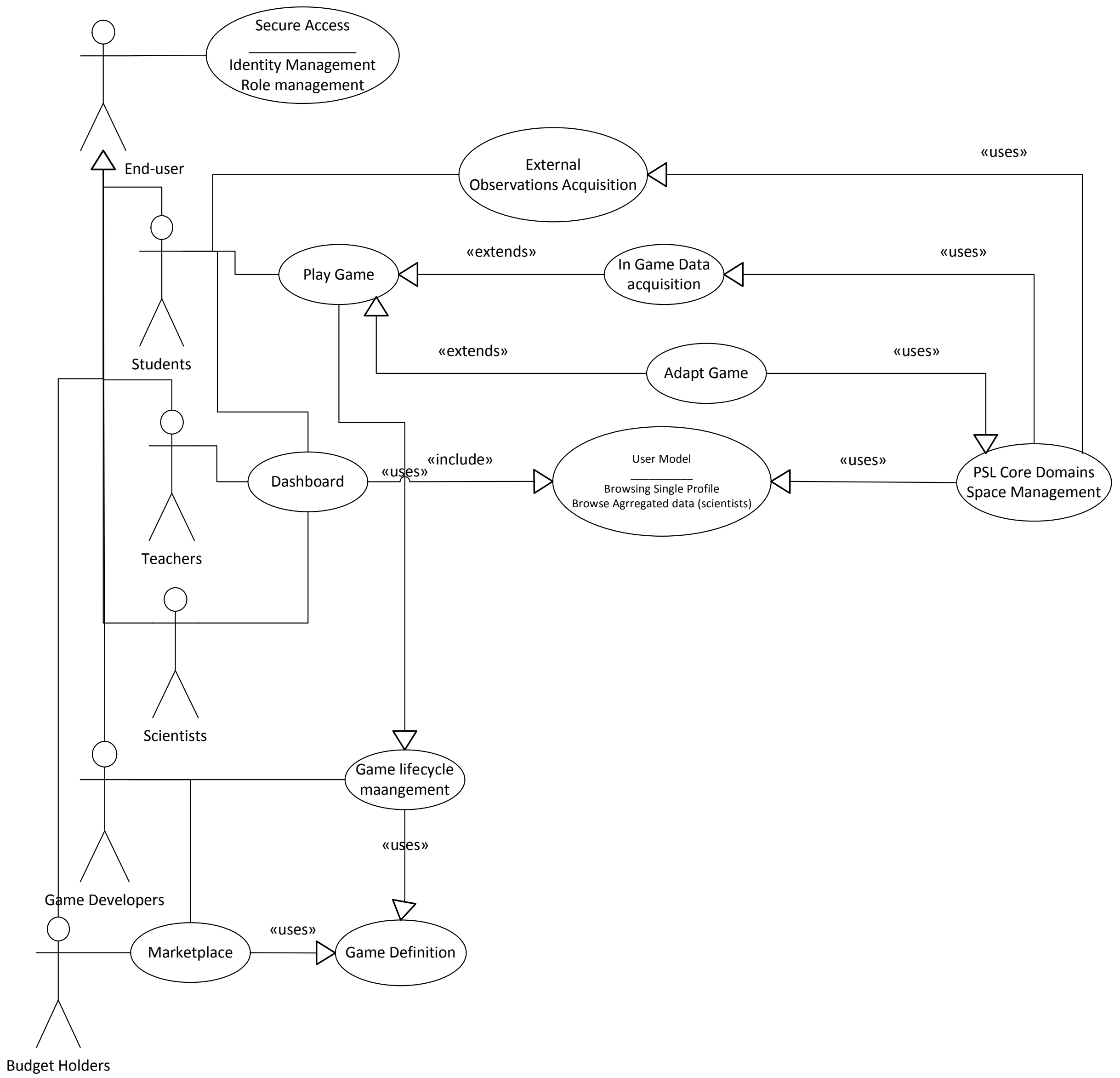

Figure 4 - Use Case diagram (level 0)

Figure 5 - Use Case diagram (level 1) shows a refined use case diagram where the main use cases have been broken down into smaller functionalities while Figure 6 - Use case diagram showing non-human actors introduces non-human actors. In the diagram it is possible to identify the “PSL Core Domain Space Management” that is one of the core services associated with the platform. The following paragraphs describe in more detail the different use cases.

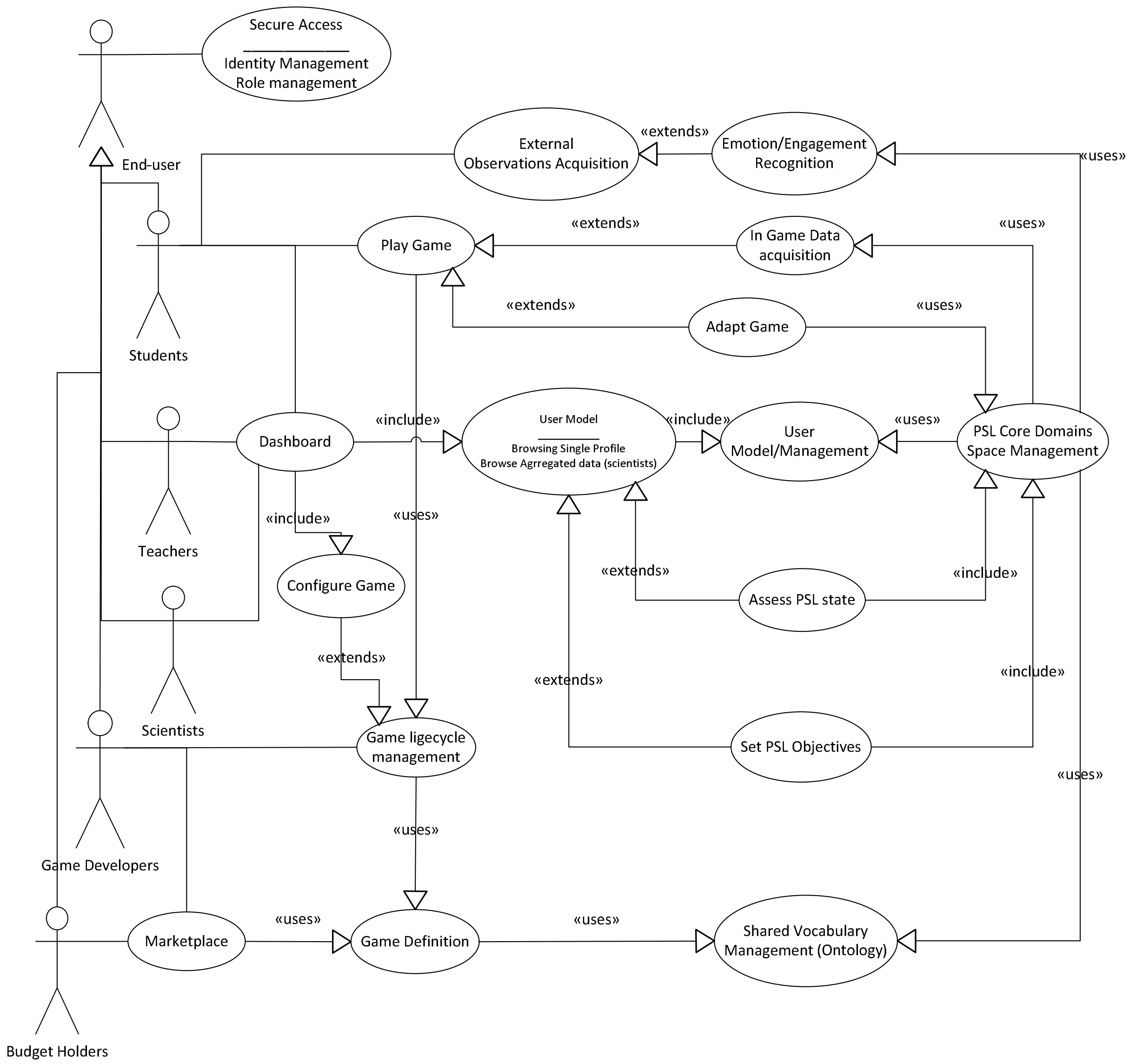

Figure 5 - Use Case diagram (level 1)

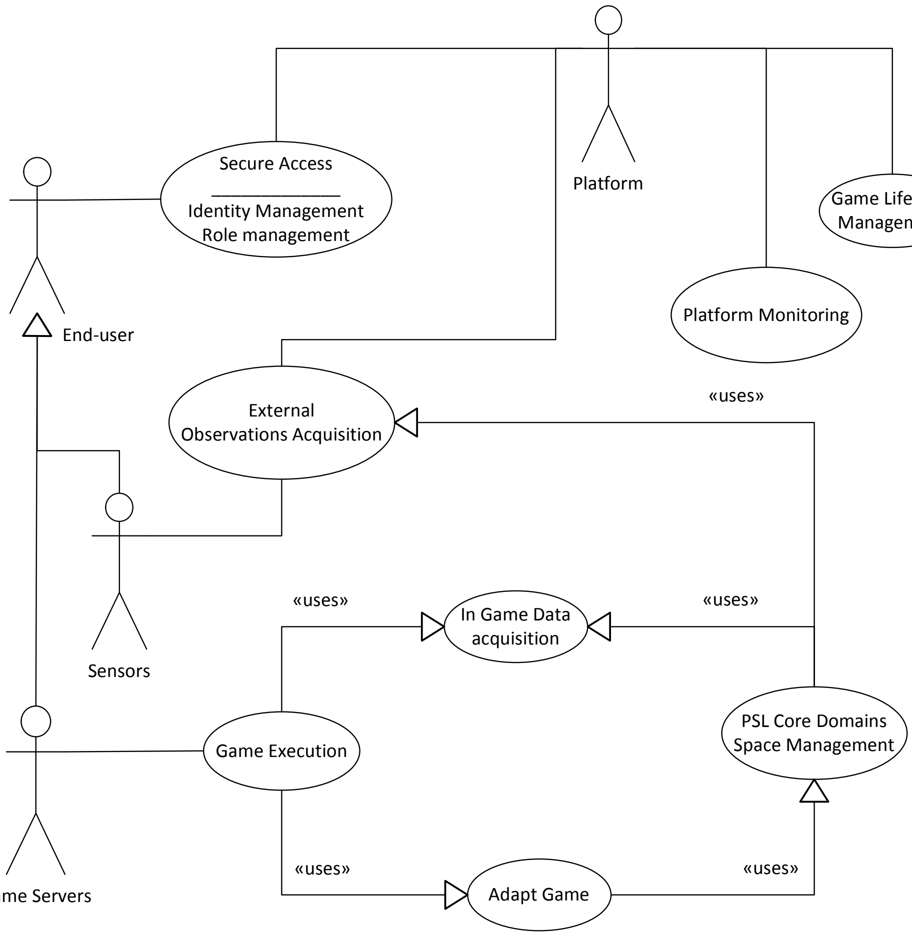

Figure 6 - Use case diagram showing non-human actors

3.3.1.1 Users

The identified users are:

- End Users. This is a generalization and describes those who access the basic service of the platform (Service Access)
- Students represent the users of games. A profile is created and associated with them. Through their profile, the behavioural observations collected and the in game data produced are the centre of Prosocial State assessment and Objective definition.
- Teachers and Scientists. These classes of users have similar interest in browsing data either for getting single profile information (teacher) or aggregated information for validating scientific hypotheses (scientists). They also can perform lightweight game configuration
- Game developers. This class of users creates games and deploys them on the platform, also assessing the services offered by the marketplace. They also need to access a service managing the vocabulary, so that games and prosocial services offered by the platform use a common grammar and share the meaning of terms and concepts.
- Budget holders. They access the marketplace choosing the games that suit them. They also need a clear game definition, so that the goal of the purchased games is clear.

- Platform (cf. Figure 6 - Use case diagram showing non-human actors). This fictional user represents the platform upon which the different use cases are realized. The relationship with the use cases should be therefore read as service offer.
- Sensors (cf. Figure 6 - Use case diagram showing non-human actors). This fictional user represents the different sensors that are used to capture user information like voice or facial expression.
- Game Servers (cf. Figure 6 - Use case diagram showing non-human actors). This fictional user represents the game server actually deployed on the ProsocialLearn platform.

3.3.1.2 Secure Access

This Use Case is associated with all the users, and represents the functionalities associated with the control of access to the platforms and the games. All the users have a role associated with them that defines their mission in the system and the Use Cases applicable. The roles will define the possibilities of the users in the platform, ensuring also the security and privacy. Role management will operate at record level (not just predicating on the schema), so enforcing a high level of privacy and making data available only to the owner and only to other users with the right role and right relationship with the owner (e.g. each teacher will see only students' data belonging to their classes).

3.3.1.3 Observation Acquisition

This use case represents the set of operations that enables the acquisition of raw observation to be interpreted mainly as emotion to be then later used for assessing the prosocial state of a player.

3.3.1.4 Play Game

This use case comprises the users playing games. In playing they produce in-game data that are filtered to reduce their size and then later used for assessing the prosocial state of a player and the following suggestion made by the platform to the game server in terms of possible game adaptation.

3.3.1.5 Emotion/Engagement Recognition

This use case represents the portion of activities performed in the platform for interpreting raw observation in terms of player emotions or player engagement.

3.3.1.6 In game data acquisition

This use case represents the acquisition of in game data. The game data need to be interpreted and stored according to a generic model that homogenises the different game specific logs.

3.3.1.7 Adapt Game

This is the use case dedicated to the collection from the PSL core domain space manager (see Section 3.3.1.11) information on the current player's prosocial state and the related prosocial learning goal. The set of functionalities included here translate the above information into suggested possible changes to the game engine.

3.3.1.8 Dashboard

This use case represents the interaction with the GUI offered by the platform where human users like students, teachers and scientists can access platform functionalities of interest to them.

3.3.1.9 User Model

This use case summarizes all the functionalities related to the creation of views (browsing) over the user model for satisfying different end user requests.

3.3.1.10 User Model Management

This use case represents the functionalities for creating and managing the user model.

3.3.1.11 PSL Core Domains Space Management

This represents the core of the platform, in fact it is responsible for the creation and management of the different measurable spaces where prosociality can be assessed (i.e. measured).

3.3.1.12 Assess PSL State

This use case shows the functionalities used for assessing where a student is in terms of prosociality.

3.3.1.13 Set PSL Objectives

This use case is about setting (by the teachers) and checking for the sake of game adaptation the objective a given student should meet in terms of prosocial abilities.

3.3.1.14 Create/Deploy Game

This use case represents the Game Developer adding a new game to the system. This includes writing a text description for adding it to the Marketplace and using the Deployer API for uploading the game files to the PaaS platform

3.3.1.15 Configure Game

In this use case the teacher/scientist uses the Dashboard for setting up the educational parameters of the game.

3.3.1.16 Game Definition

This use case represents the functionalities used to fully describe a game in terms of the prosocial features it supports. It enables the possibility of a teacher (or a budget holder) to choose the best suitable game for a given purpose. The use case also includes the technical specification of the game so that the platform knows how to handle the data it produces to interpret them in prosocial terms.

3.3.1.17 Shared Vocabulary management (Ontology)

The mutual understanding of platform and games is fundamental for operating them. An ontology is produced to describe and store all the concepts and their mutual relationship. Specifically a vocabulary is offered for an easy support to understand the definition of terms and concepts as they are used in the platform.

3.3.1.18 Marketplace

This Use Case represents the Budget Holder choosing and acquiring games that suit the educational needs.

3.3.1.19 Game Execution

This use case (cf. Figure 6 - Use case diagram showing non-human actors) represents the dual of "Play game" in Figure 5 - Use Case diagram (level 1) where the focus is on the functionalities necessary for running a game.

3.3.1.20 Platform monitoring

This use case (cf.Figure 6 - Use case diagram showing non-human actors) represents the monitoring functionalities that the platform implements for controlling the status of the game. It is the base of the support for scalability offered by the platform.

3.4 Information System architecture

Following the TOGAF methodology the information system architecture is broken down into Data and Application System architecture. The former presents the relationship between data and the latter presents the different software modules.

3.4.1 Data architecture

This section is organised as follow: first we introduce how we represent observations and the relationship among data, and then we move to describe the principal sets of data models. Not all the models have the same maturity because they are going to be further specified and developed later on in the project (they will be reported in associated dedicated deliverables). We list and describe all the models in this deliverable so that the global picture of the ProsocialLearn approach can be appreciated. The principal set of data models is enumerated below:

- Prosocial Core Ontology
- Game specific run-time model
- Emotion/Engagement models
- Prosocial measurable space model
- User model
- Prosocial evaluation and adaptation model

3.4.1.1 Observation Modelling and relationship among data

Many different data, generated from observations and machine processing, are part of the PSL ecosystem and all of them need to be properly modelled and then managed. Observation and modelling plays a pivotal role in the overall functioning of the PSL architecture. Observations drive experimental insight since it is through the understanding of the relationships between prosocial related activity and emotional responses that we develop improvements to prosocial game design and adaptation. At run-time, observations of the human player's emotions, psychological characteristics and in-game related activities are used to capture evidence of 'game transactions' (described later in this document) that lead to the identification of prosocial behaviours.

Multiple stakeholder views from within the PSL ecosystem demand a structured and also flexible, model-based approach to handling data streams (from multiple sources) that are attributed to entities having dynamic aspects and life-times within the environment. For example, teachers will wish to understand student learning achievements as they relate to prosocial learning objectives (PLOs) through game-play. These achievements need to be evaluated in terms of the interactions and emotions that players exchange with one another and other game entities at run-time. Game developers, on the other hand, will want to know how to map their specific game logic to descriptions of prosocial related activities such that game adaptation recommendations (to enhance student learning) can be generated. Since it is undesirable to restrict 3rd party game developers to a narrowly defined set of game genres and mechanics, the PSL data architecture must offer a framework of prosocial semantics that are aligned and easy to extend to high-level game design concepts.

In the PSL architecture observations and models of human and game behaviours are closely related and for this reason a model-based approach to observation is adopted. We build upon the observational framework provided in the EXPERIMEDIA project[10,11] in which targets of observation are formally modelled as entities that have both metric and semantic data channels associated with them. Quality of service (QoS) and quality of experience (QoE) metric definitions and measurement sets are supported in a metric oriented data model, as described within EXPERIMEDIA's EXPERImonitor API[12]; see the figure below.

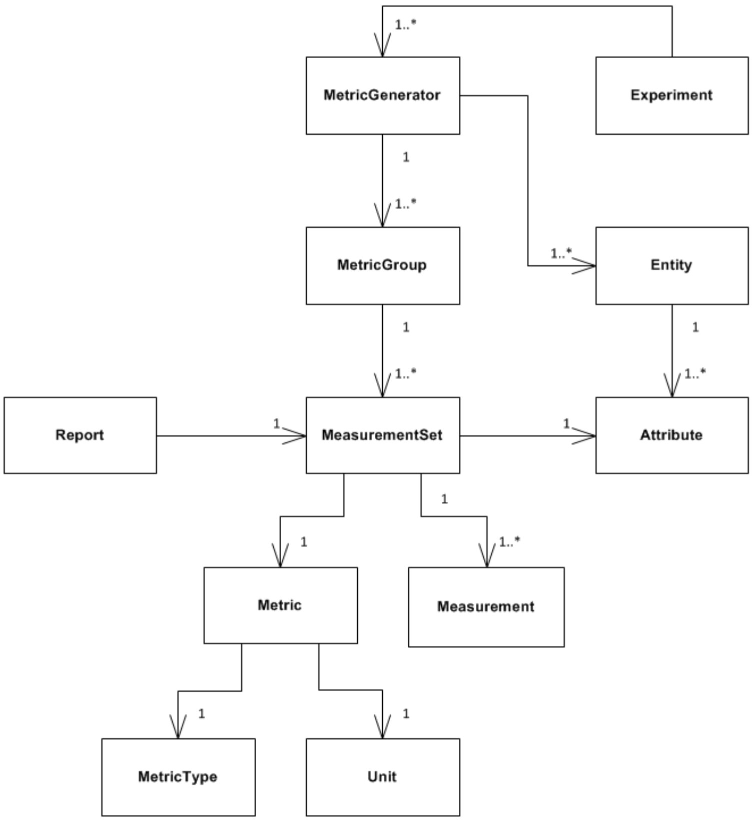

Figure 7 - EXPERImonitor metric data model

In this metric oriented view, entities (subjects of observation) are uniquely identified individuals that have one or more attributes (specific features that will be observed). Metric generators, associated with a given experiment, contain logical groups of metrics that represent sets of measurements related to a particular attribute. Each measurement set has a metric meta-data associated with it, provided by its parent measurement set, including type and unit of measurement. Measurements are provided to the EXPERImonitor experiment management service via report records. For more details about this model, please see the EXPERImonitor documentation[12].

[10] http://www.experimedia.eu/

[11] S. C. Phillips, Stephen C., M. Bashevoy, M. Boniface, S. Crowle, V. Engen, Z. Sinkala, and S. Wiegand. Linking quality of service and experience in distributed multimedia systems using PROV semantics. In, IEEE 9th Int. Symposium on Service Oriented System Engineering (SOSE), San Francisco, US, 30 Mar - 03 Apr 2015. IEEE.

[12] https://github.com/it-innovation/EXPERImonitor

Additional semantic data can be linked to experimental entities through EXPERImonitor's support for RDF based triple data. Human and system behaviours (which may be linked to metric entities) are represented as W3C OWL[13] based class types and augmented with the provenance ontology specified by the W3C Provenance[14] standard.

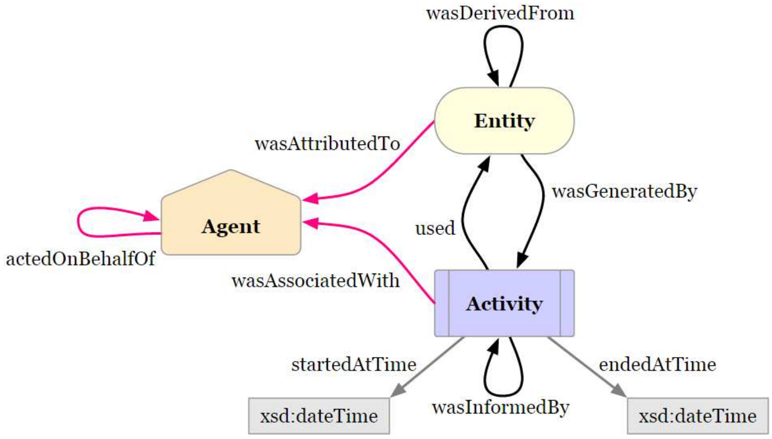

Figure 8 - W3C Provenance Core types[15]

The EXPERImonitor API supports the exploration of model based observations by offering a time based view of unfolding human and system behaviours using the W3C Provenance based constructs to navigate through complex, graph based interactions between entities, activities and agents. In the PSL architecture we will exploit and extend this approach to observation and modelling by introducing new prosocial based ontologies that will provide new capabilities allowing game transactions to be identified and reasoned with at run-time both by humans and non-human systems.

3.4.1.2 Prosocial core ontology

This ontology is provided as a means of realising a common, semantics based foundation for describing human and computer behaviours that are of concern to project stakeholders with a direct interest relating to game design and learning. The complete ontology model is presented in the figure below.

[13] http://www.w3.org/standards/techs/owl#w3c_all

[14] http://www.w3.org/TR/prov-o/

[15] Diagram provided by W3C, see http://www.w3.org/TR/prov-o/#description

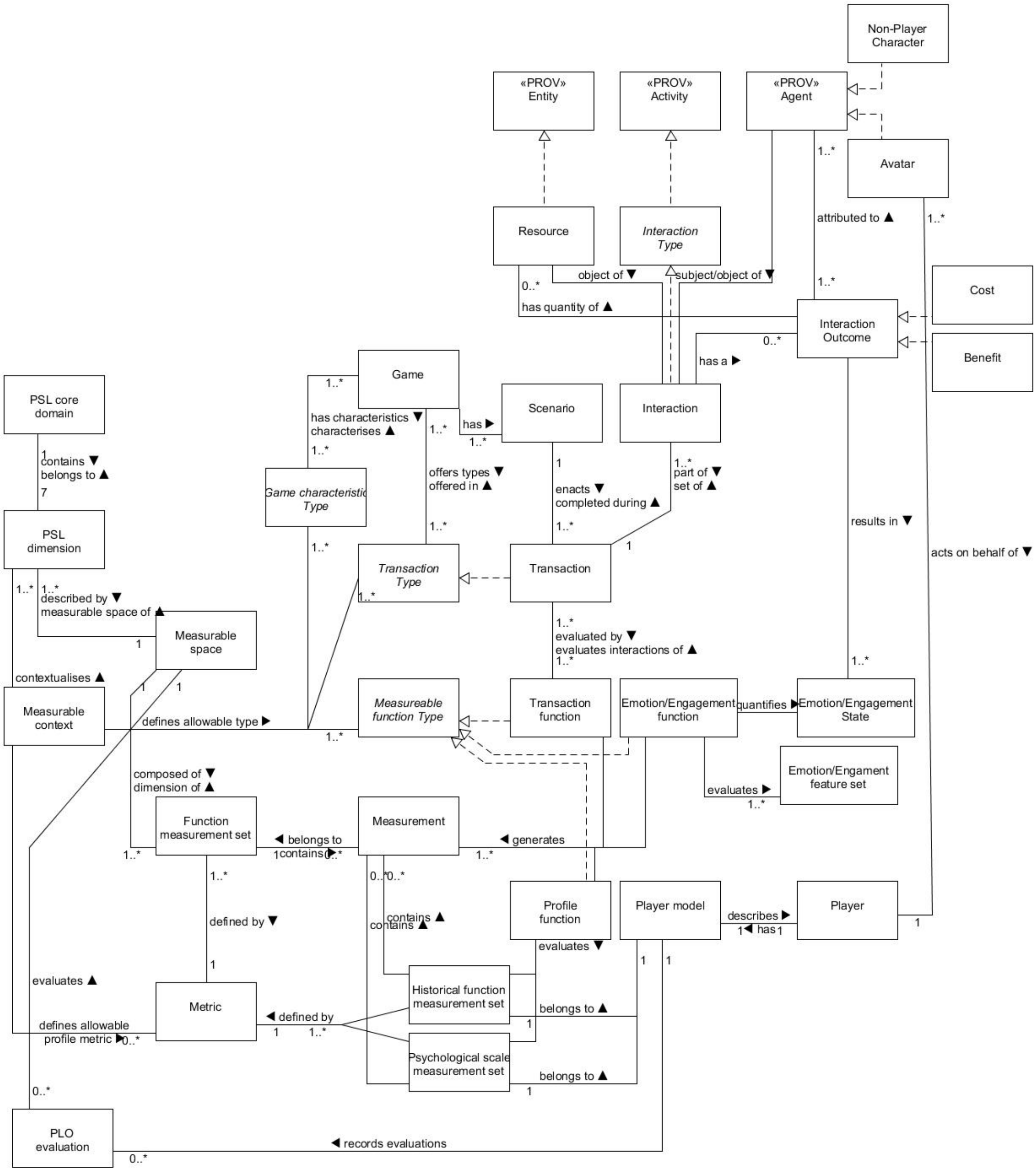

Figure 9 - ProSociaLearn Ontology

Rather than deconstruct the ontology directly from the figure above, we will present smaller views of the whole in the following sections:

- High level game concepts specification
- Game transaction specification
- Prosocial measurable space specification
- PLO region definition

3.4.1.3 High level game concepts specification

We begin with a subset of the core ontology that offers a high level representation of game concepts and structures. This model and the run-time data derived from it has been defined for two reasons: a) to offer game developers recognisable types that can be mapped to their own game logic and b) to provide a structural backbone for use in exploring and navigating through more complex prosocial observational data.

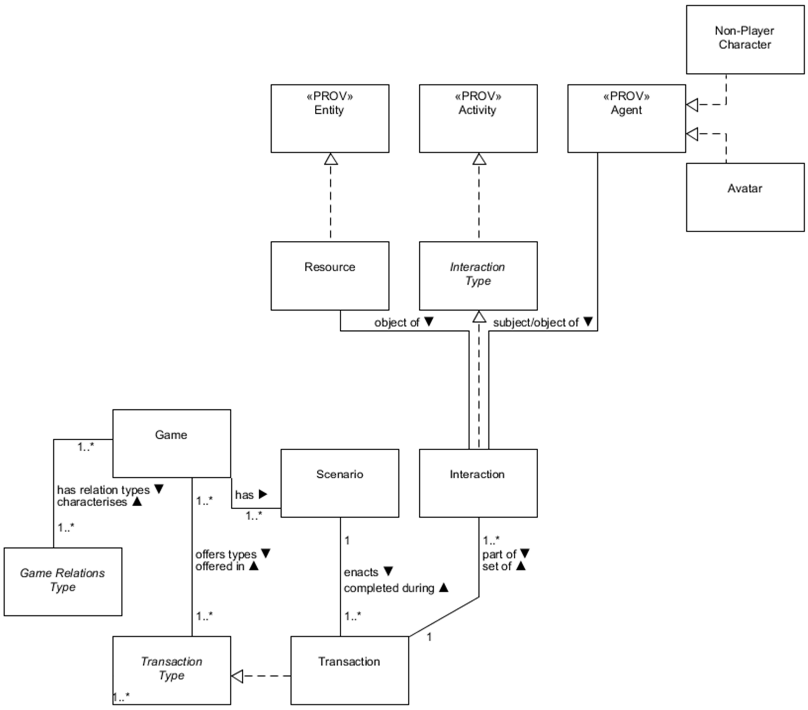

Figure 10 - High level game concepts

In Figure 10 - High level game concepts above we present these high-level game concepts. A *game* is considered to have a number of *game relation types* that describe aspects of a player's relationships that can be formed with one another, such as whether the game is friendly or combative; anonymous or social. These elements provide important information that is used in defining *contextual constraints* in a *prosocial space*. Games are composed of one or more *scenarios*; these provide a simple structural framework that can be used to logically group types of game *transaction* as well as creating a possible mapping to typical game structures such as 'levels' or 'chapters' in game narrative. Within each scenario, players of the game may take part in a number of *interactions* that are aggregated and matched against known patterns of interactions that represent *transaction types*. This aggregation of transactions provides the game with a portfolio of transaction types that characterise the game from a prosocial perspective – this mechanism is described later. Interactions themselves are typed based on the W3C Provenance *activity* class and describe specific relations that connect types of *agents* (a W3C Provenance term) with types of game *resources* (which are a sub-type of the W3C Provenance *entity* type). We derive two types of agent in the ontology: *i)* the *avatar* (that acts on behalf of a human player) and *ii)* a *non-player character* (NPC) whose actions are driven by an artificial intelligence. An example of two simple, related interactions can be described as a) '*Alice's avatar* [avatar] *gives* [interaction] *sweets* [resource]' and b) '*Bob's avatar* [avatar] *receives* [interaction] *sweets* [resource]'.

3.4.1.4 Game transaction specification

Game transactions are collections of interactions that have been matched to particular aspects of a prosocial behaviour (an example of this is provided in Section 3.4.1.6). Each interaction having relationships with agents and resources (described above) also has zero or more *interaction outcomes*. An outcome has a potential relationship with a resource through some quantifiable expression (such as a percentage or a count of the resource) or a value representing an emotional state. Agents representing either human players or NPCs are attributed to an outcome. Finally, outcomes may be interpreted as either costs or benefits to an agent using the associated relationships. It is anticipated that in many cases a transaction will represent a series of interactions that have outcomes consisting of both costs and benefits for a number of agents. A transaction function will use this data to generate a metric relating to a prosocial behavior.

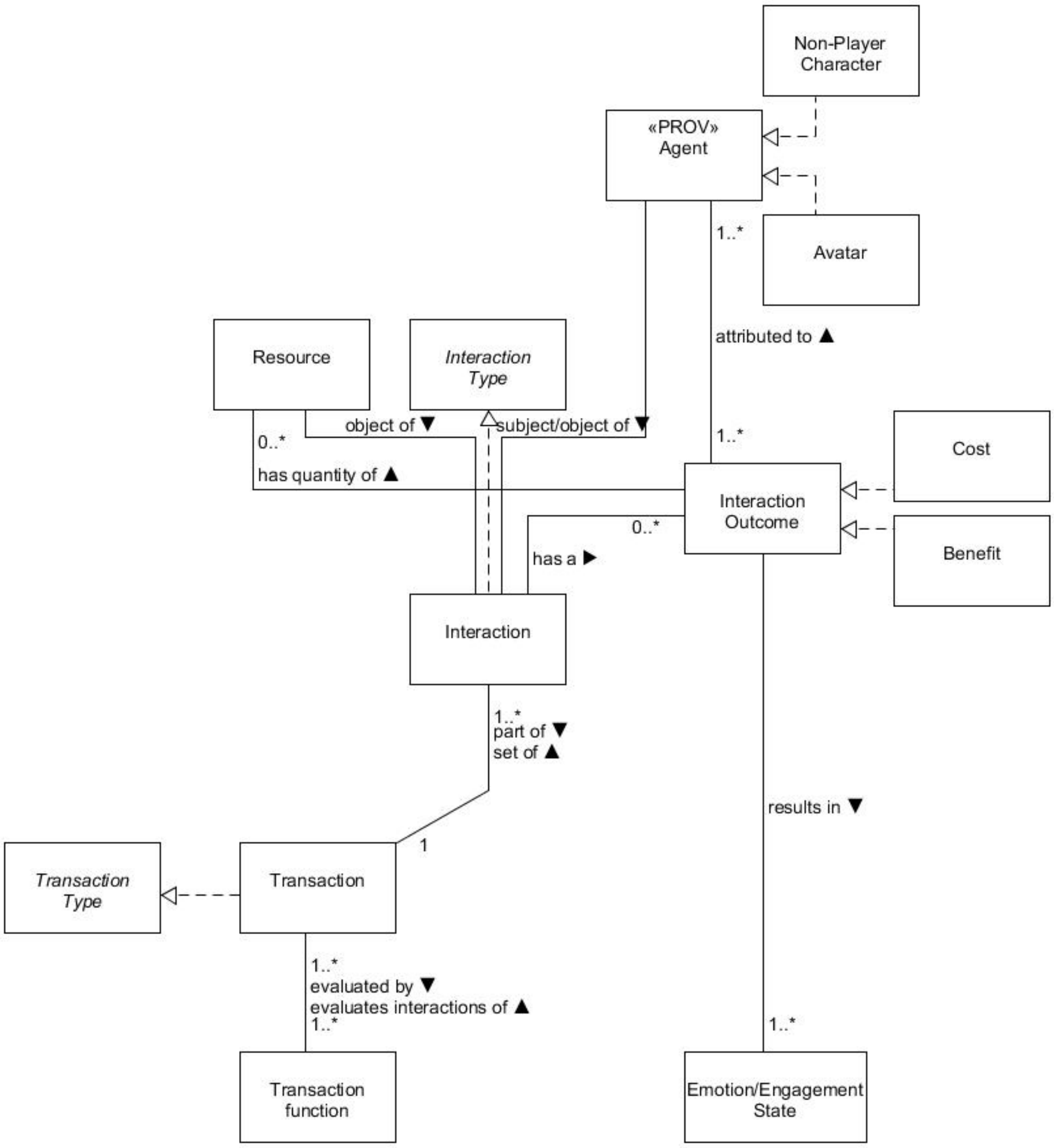

Figure 11 - Game Transaction Specification

Prosocial measureable space specification

The prosocial measureable space (Figure 11 - Game Transaction Specification) is a multi-dimensional space structured by orthogonal axes that describe quantitative metrics generated by functions that work on observations of human emotion, in-game transactions and data from the player profile. A holistic view on the application of this space for understanding a learning and development in a prosocial domain is given later in this document in Section 3.4.1.6.

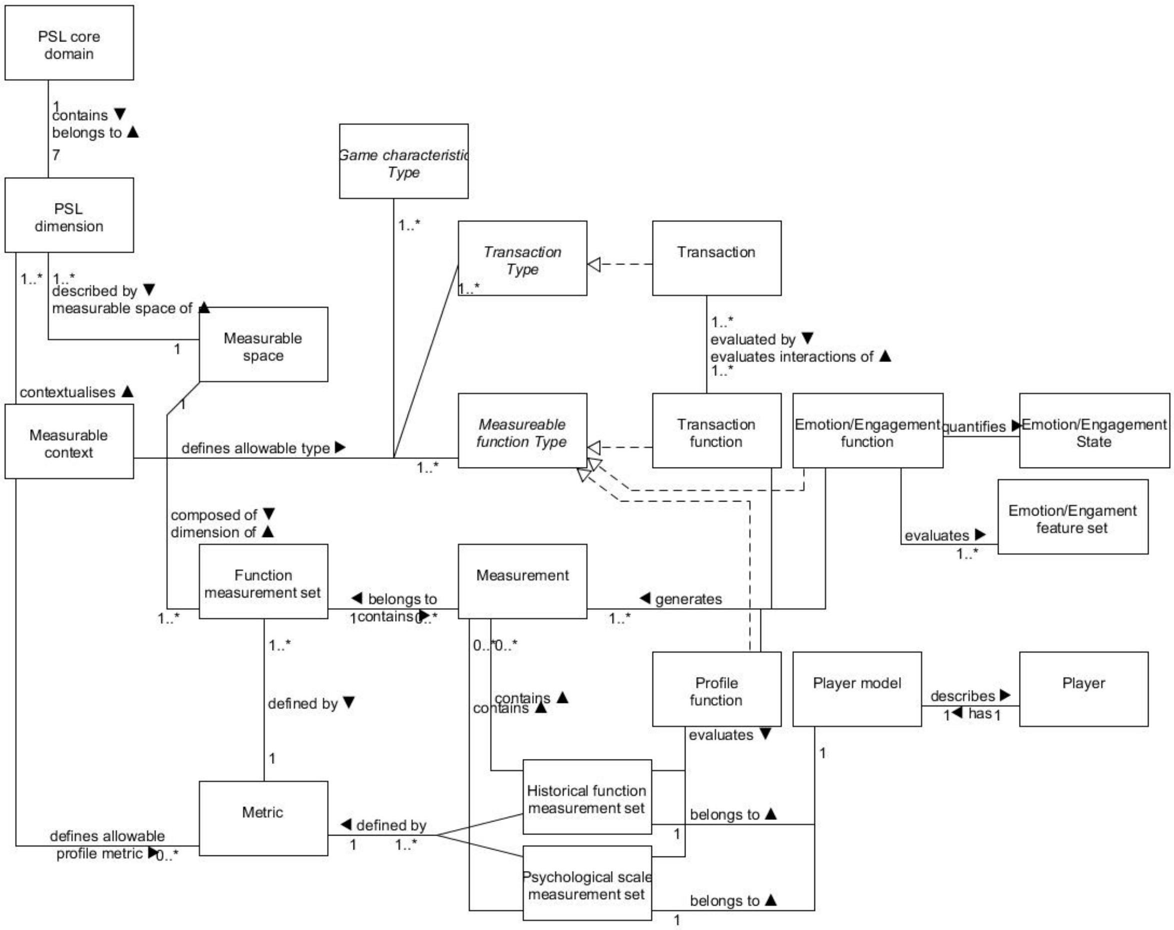

Figure 12 - Prosocial Measurable Space definition

In Figure 12 above we present the semantic arrangement of components that specify the prosocial measurable space. As already determined in other ProsocialLearn deliverables, the PSL core domain consists of seven elective dimensions (see D2.1 for further information). Each dimension has one or more *measurable spaces* associated with it that is composed of metric based sets of measurement – one for each dimension of the space. Each metric has a description of the type of numerical quantification used and unit associated with it (based on the EXPERImonitor metric data model). Data that describes some value for each dimensional axis is generated by a function which has a type of either *transaction*, *emotion* or *profile*. Transaction typed functions use data collected from in-game transactions (such as quantifiable values related to *costs* and *benefits*) to arrive at a metric based value relating to some aspect of prosociality. Emotion “functions” derive a metric based value related to a specific emotional state based on computation over feature data captured from observations of the human player. Profile functions generate metric based values derived from historical records of transaction or emotional functions or other prosocial data captured from psychological questionnaire data.

"Measurable spaces" are explicitly qualified by a *measureable context* that is defined by a prosocial design expert and limits the allowable types of *measurable function*, *game transaction* and *game characteristic*. This qualification is important since prosocial behaviours should always be observed and understood within specific contexts that may influence the outcome of a game. The PSL core ontology describes this delimitation of the measureable space through constraints relating to the presence of specific allowable types – this ultimately constrains the available measurement sets (and therefore metrics) that can be used to describe any given prosocial measurable space.

PLO region definition

Prosocial learning objectives (PLOs) are an intrinsic part of the pedagogic process supported by the PSL project and of special interest to project stakeholders representing the teaching community. In the PSL data model, a student's progress toward a learning objective is defined through the evaluation of metrics related to a measurable prosocial space.

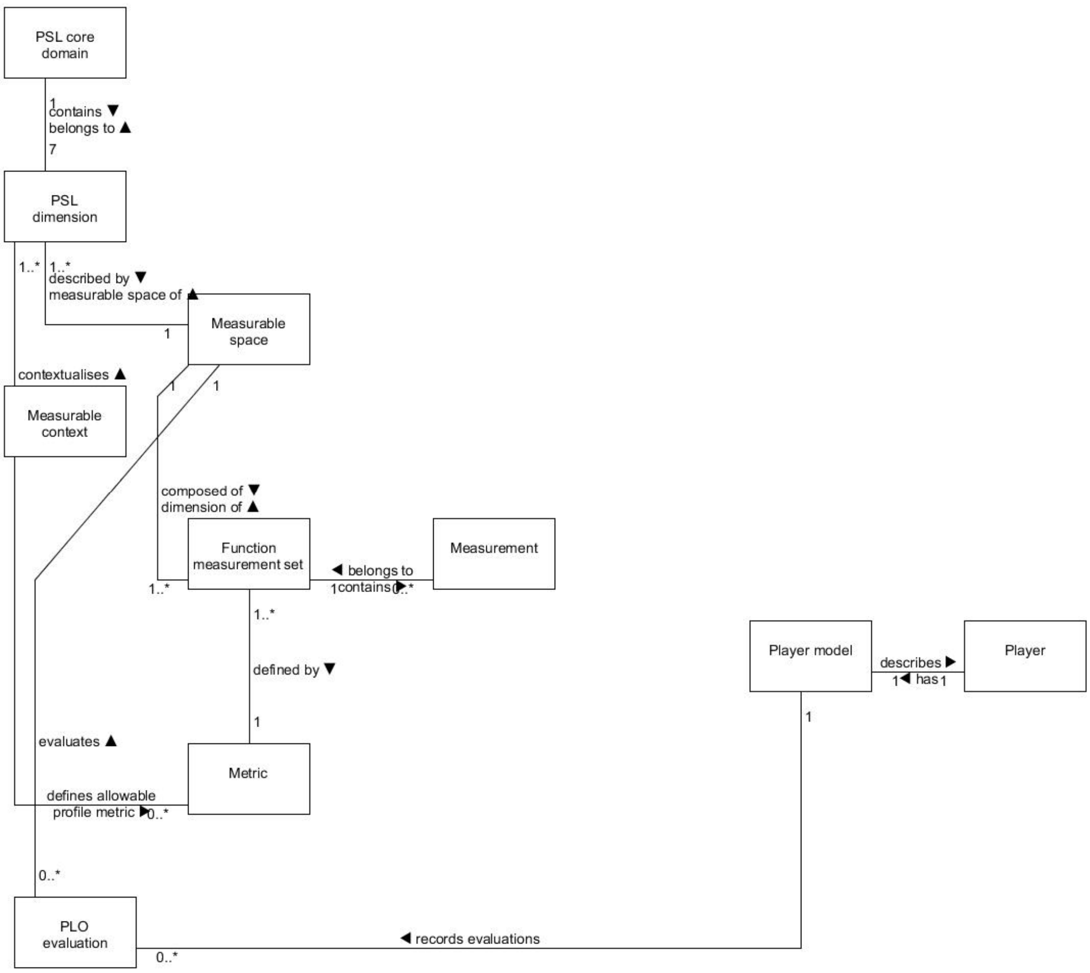

Figure 13 - PLO region definition

In Figure 13 above, a *PLO region* (bottom left) is defined as a zone in the prosocial measureable space related to a specific PLO. As games are played, player's transitions that result in them 'visiting' these regions in the measureable space are recorded in their profile. See again in Section 3.4.1.6 for an example describing the possible movement in the "fairness space".

Game specific run-time model

Developers of prosocial games will need to map their 'concrete' game concepts to the PSL core ontology by carefully selecting appropriate elements from their game design and encoding them as derivative types. This will allow the platform to correctly interpret the information coming from the game. This section is therefore a placeholder highlighting that, by using the model described in Section 3.4.1.1, a model representing the game transactions used for assessing prosociality is needed. This can be achieved either by fixing a vocabulary and a format to which all the game logs need to adhere or as a binding between the game log format and the prosocial learning concepts. In the annex an exercise is presented showing how to model one of the games the project is developing.

3.4.1.5 Emotional/Engagement models

By externally observing how the player is playing (e.g. through a webcam, or recording her voice), it is possible to extract useful information on the emotional state of the player and also on their engagement. Both these aspects are relevant for assessing the prosociality of a student. The models will be described in D3.1 and here we sketch the main characteristics in terms of input and output.

Many types of observations will be supported in the project. Currently works are in place around facial expression, voice, and gaze analysis. All the analysis will be based on data fusion techniques (discussed in Section 3.4.2.7) that will operate at different levels. First features will be extracted that are relevant for the analysis. Then a classification of the extracted feature will be performed by applying machine learning techniques. This will lead to an assessment of a component on a single modality fashion (e.g. the user's emotion through voice). Finally the results of different modalities will be combined providing a multi-modal assessment. From a data model point of view it is therefore necessary to define several of them. For each supported modality a "low level" data model will describe the extracted features (e.g. voice frequency), while a high level data model will describe the high level output (i.e. emotion or engagement). These models will be numeric and will include error margins.

3.4.1.6 Prosocial space model

Operationalising psychological frameworks is quite complex work discussed for instance by Bar-Tal in 1986[16], and more recently by Wentzel[17], and by Wyeth[18]. The goal is to define computational spaces (not necessarily in a strict mathematical sense) where psychological aspects can be measured. In the case of ProSocialLearn the measurement needs also to be as much automatic as possible.

[16] Bar-Tal, D. (1986). Altruistic motivation to help: Definition, utility and operationalization. *Humboldt Journal of Social Relations*, 3-14.

[17] Wentzel, K. R., Filisetti, L., & Looney, L. (2007). Adolescent prosocial behavior: The role of self-processes and contextual cues. Child development, 78(3), 895-910. doi: 10.1016/0022-1031(83)90025-2

[18] Wyeth, P., Johnson, D. M., & Sweetser, P. (2012). Conceptualising, operationalising and measuring the player experience in videogames. Paper presented at the Extended Proceedings of the Fun and Games Conference 2012.

The prosocial domain space model is a framework within which the PSL core domains are related to *measureable spaces* representing the prosocial behaviour of players. A PSL core domain may be characterised by more than one measureable space since the definition of a prosocial behaviour is in part characterised by the context in which it occurs. For example, the degree to which a player's actions may be considered 'fair' may have variable influencing factors, such as when he or she knows the identity of the other player(s). In another case, the player's predisposition to act fairly (determined by a previously calculated metric based on psychological scores or behaviour from game history data) may also affect the overall evaluation of fairness in a specific case.

In Figure 14 - Example of Fairness measurement below we illustrate an example of a measurable space relating to fairness described using three simple metrics: a) an emotional scale (happy to sad, along the x axis); b) a fairness score (0 to 5 inverted along the y axis); and c) time taken to perform the related game transaction.

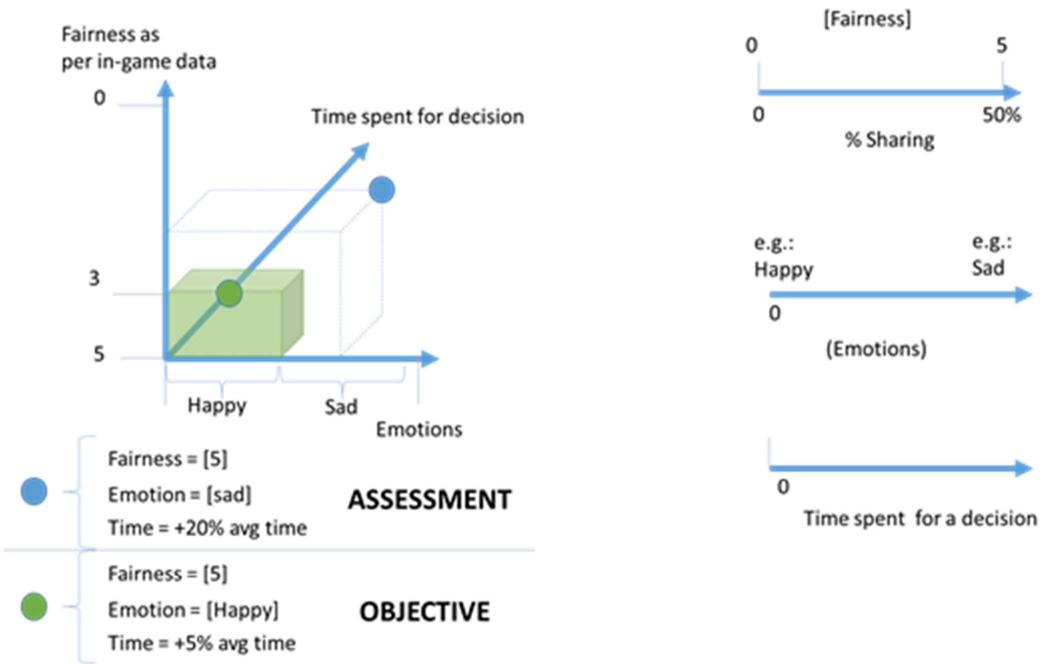

Figure 14 - Example of Fairness measurement

At run-time, data captured from observations of face and voice features provide the basis for the emotion metric expressed along the x axis. Similarly, game transaction data captured from observations of in-game interactions (originated from logical game logging) will be used in the calculation of a fairness score (found on the y axis) and also used in the calculation of the time taken to complete a fairness related game transaction. An assessment of a player's learning can be described as a *transition* from one point in this measureable space to another over the course of one or more games. Some region of the measurable space (indicated in green in the figure above) is considered as a target for the player in which the combination of game transactions and emotional responses is considered sufficient to meet a prosocial learning objective (PLO).

The dimensions of a measurable space are defined by a *measurable context* that is a mapping between a set of *constraints* describing specific user and game characteristics and a particular PSL core domain. As a group, these constraints are used to qualify the applicable *evaluation functions* available to generate metrics that make up a measureable space. These relationships are illustrated in the figure below in which a specific measureable space of PSL core ‘fairness’, referred to as *Vector A*, is linked to a set of contextual constraints that lead to the identification of relevant prosocial transactions and evaluation functions.

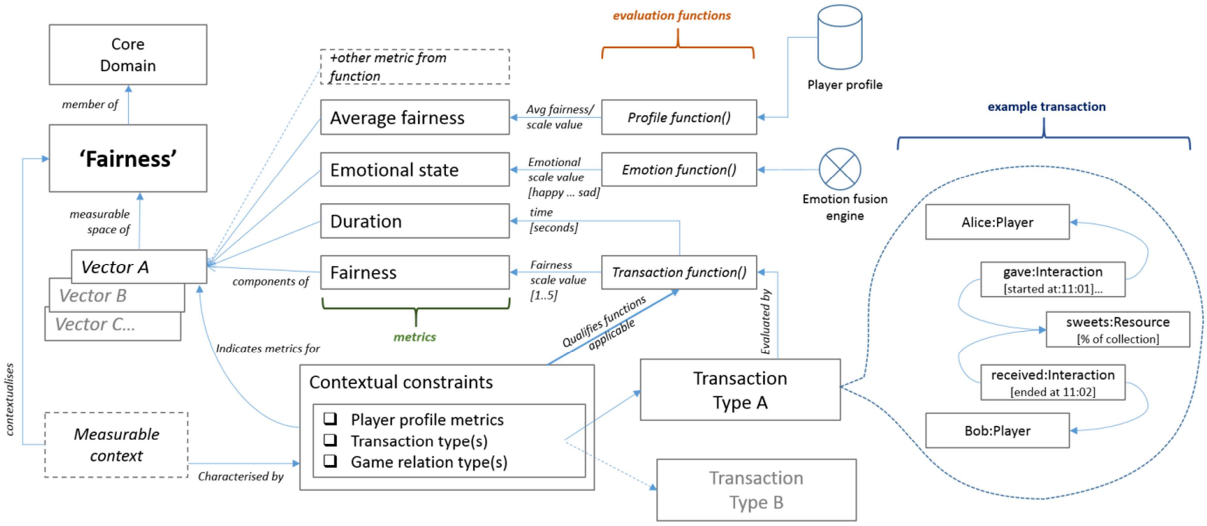

Figure 15 - Contextual constraints and metrics used to define a measureable space for the 'fairness' core domain

Contextual constraints are representative of a number of characteristics that relate to both player and the game relationships possible. Characteristics relevant to the user are related to metrics previously generated by profile functions that work on data gathered from either psychology questionnaire scale data or metrics generated from previous prosocial gaming sessions. In this case, this metric would represent a dimension in the measureable space that has a single, constant value. Dimensionality of the measurable space defined by these elements is optional.

Further description of the measurable space is determined by the definition of prosocial transactions allowable in the measureable context and, by extension, the set (or sub-set) of applicable *transaction functions* associated with it. Transaction functions generate metrics based on an evaluation of the interactions (and their properties) within the scope of a game transaction; in the example above we see the transaction function for ‘share transaction’ generates a fairness value and a duration. To elaborate further: in this simple case, the property of the resource element of the transaction (a percentage of sweets given by Alice) is mapped to a fairness value given somewhere in the range of 1 to 5. The duration of the transaction is computed using the time stamps attributed to the ‘give’ and ‘receive’ interactions. It should be noted that any number or type of prosocial transaction can be included in the contextual constraint – leading to a measurable space that could include more than one core domain related metric. Finally, the emotional state metric (a scaled, bi-polar value) is derived from an emotion function that operates on the classifications and confidences (related to the player) generated by the emotion fusion engine. Again, the dimensionality of the measurable space defined by emotion based metrics is optional. It should be noted that the model works under the assumption that it is possible to have independent measurement of in game data and emotion and that the known relationship between them is intrinsically captured them through the observation of the admissible points in the space. In other words a change in the emotion may well have an impact in the game mechanism. There is not a direct formula for representing this influence that is captured by directly observing the outcome across the different dimensions.

3.4.1.7 Player profile

The PSL data architecture encompasses multiple aspects of the player which include aspects of them as a game player and a prosocial learner. In Figure 16 below, these are summarised in a class diagram.

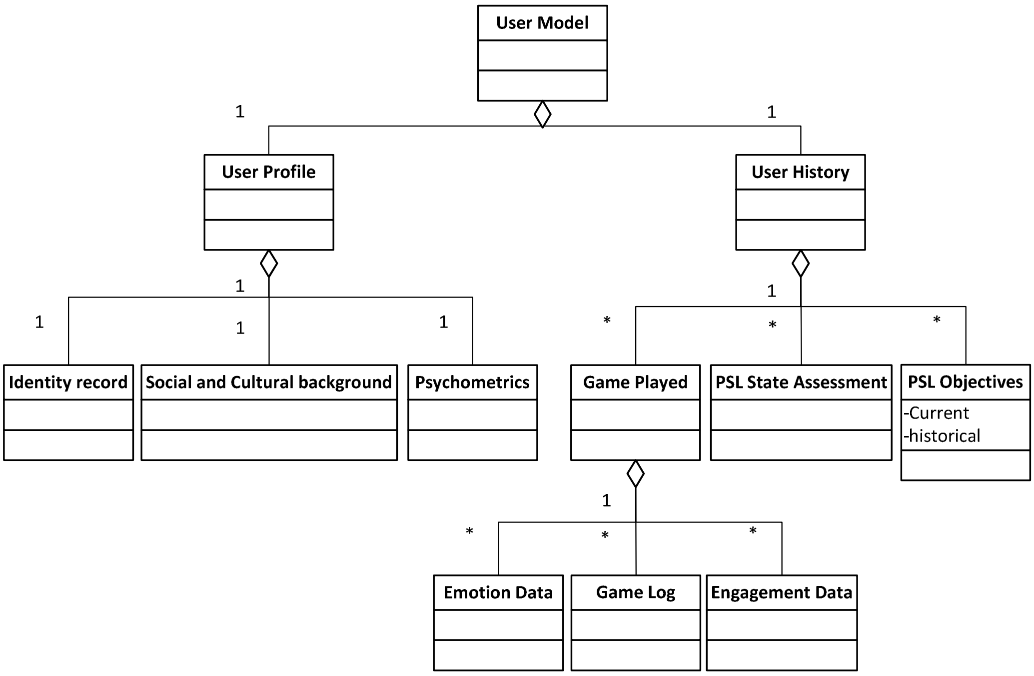

Figure 16 - User model

The user model needs to be built across different runs of the same game and/or across different games, but, at the same time, the identity record should be always kept safe and separate from the rest of the data, so that outside the platform it is impossible to link sensitive information to specific identity. The model and the associated engine should therefore support privacy-enabled queries where the returned dataset satisfy the associated privacy rules. The user model is divided into two parts, one modelling information obtained while the player is playing (right part of the diagram, called User History) and one modelling information obtained externally via other means, such as questionnaires. The identity record information are kept separately for privacy and security reasons and the player is identified in the platform by a unique identifier, traceable back to the real identity only by the platform for providing persistence across games and by the users (like teachers) with the right to access such information. Other information stored in the user profile are related to cultural, societal and psychometric characteristics, but also simpler information like the language spoken, so that the game can be provided to them in the right language.

Games that a user has played will be captured in their game history record in which a list of games and time played will be maintained. Orthogonal to this temporal view of game play, a set of PLO achievements will be managed for each player; as players attain new PLOs through game play, these will be added to their personal history. Underlying the temporal and PLO achievement records is a representation of the user's prosocial behaviour as represented as transitions between locations in the core domain spaces. This record provides a more technical view of the student's prosocial behaviour (as it is modelled in the project) and is anticipated to be useful in experimental and testing scenarios.

3.4.1.8 Prosocial evaluation and adaptation model

Evaluation and adaptation are important summative functions of the overall learning and gaming experience supported by the PSL architecture. Both students and teachers benefit from feedback during the learning process that is based on the evaluation of prosociality at run-time. This evaluation is realised in the form of assessing a student's transition through a prosocial space (marked by a succession of game transactions, emotional responses and engagement level) and matching it to regions in the space that are recognized as matching a PLO achievement. It should be noted that attainment of a PLO is expected to be qualified in some part by the student's psychological profile and the contextual constraints of a particular game. These factors may *widen* or *narrow* the possible transactions and emotional dimensions that allow the student to transition to a part of the prosocial space that is recognisably a progression towards understanding prosocial behaviour. This model is the way in which the platform suggests to the game engine modifications in order to fulfil a prosocial objective. The model needs to be generic and the suggestion will be predicated on the shared ProSocial model, also including possible game mechanics to use to implement the modifications. D4.1 due on M15 is in charge of fully describing this model.

Because games and platform are decoupled, it is not suitable to foresee any specific imposition from the platform to the game. Adaptations will be suggested by the platform to the game by means of a common shared vocabulary valid for all the games (e.g. "please increase difficulty in the game"), with the game developer left to decide if and how to implement the suggestion. Adaptation mechanism will be linked to the prosocial measurable space, such that, if implemented, the result should be a transition performed by the student in such a space leading him towards his defined PLO.

It is possible to anticipate that the system will support two adaptation mechanisms:

- Off-line adaptation, which affects the setup of the game environment before gameplay begins; and
- On-line adaptation, which affects the course of the game in real-time.

More specifically, the off-line adaptation will consider historical player profiles and specific constraints of PLOs. The main objectives of this mechanism are to initialize game-related parameters for optimized learning and engagement as well as to define the initial parameters of the on-line adaptation mechanism.

On the other hand, the on-line adaptation mechanism will deliver suggestions in real-time based on artificial intelligence algorithms. The on-line adaptation mechanism will consider the player's current prosocial state (may be limited to engagement at the beginning) as well as gameplay information and player data (through off-line adaptation) aiming to affect the ongoing course of the gameplay. More specifically, the adaptation mechanism will suggest a specific set of mechanics in order to either inject the proper amount of challenge to the game to optimize the level of engagement or improve the player's prosocial behaviour within a predefined prosocial space, i.e., the adaptation mechanism will change the game towards the maximization of the player's prosociality index.

3.4.2 Application Systems

In TOGAF terms the application systems represent the set of software application that are used in the architecture. In our scenario it can be seen as the description of the software modules composing the platform. Figure 17 shows the component diagram. The darker striped modules are hosted in the platform, but from an architectural point of view they are external. The game controller module is effectively optional and represents the case in which sensors (like motion detection) are used to control the game.

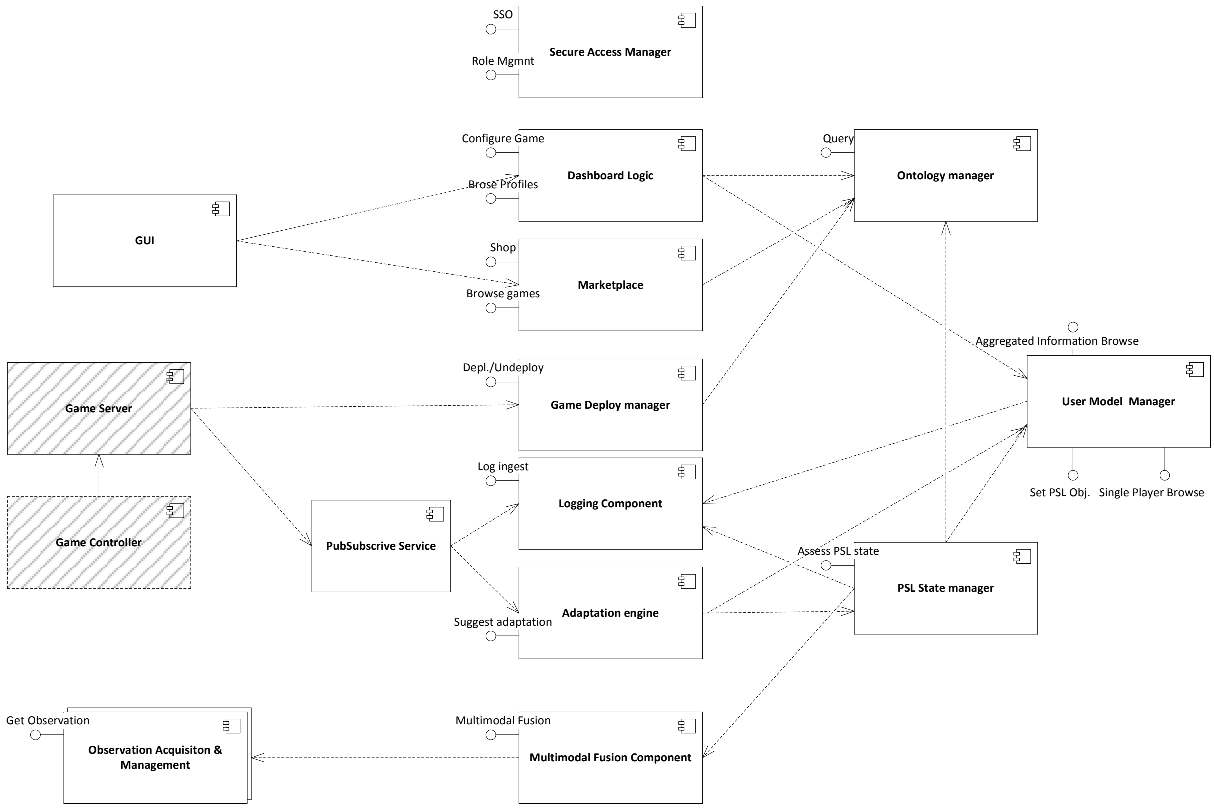

Figure 17 - Component Diagram

3.4.2.1 Secure Access Manager

The Security Access Manager is the component that provides the identity and privacy information. The Security Access Manager provides the Single Sign On information to the other components. This information is used by the platform itself, and also it is important a unique identifier for each user in the games, because the system needs to compare and summarize the performance of the players in different games.

Access to the system is related to the business side. One of the missions of the Secure Access Manager is to control (according to the policies and billing state) which users can access the system and games.

Another responsibility of the Secure Access Manager is to store and manage the information about the roles in the system. Some information in the system must be restricted only for some roles. For instance, only users with the role of teachers /scientific should be able to read the some information about the progress in the Dashboard.

But not only roles need to be taken into account. For instance, teachers should be able to see the profile information, but only about their own pupils.

Several steps will be taken in order to protect the privacy of sensitive information. The first principle is to store information which is not sensitive if possible. In general, information must be anonymized, and separate databases must be used. The Secure Access Manager will be the only component able to access the information in the database of identity data.

3.4.2.2 Dashboard

This module will comprise a GUI accessible by the end-user and a back-end module in charge of supporting the GUI providing the right data. The adherence to the MVC pattern will support the separation between presentation and back-end logic including data access. It is important to understand that the module will be used by different stakeholders and that the platform needs to intrinsically support their different viewpoints. This component will be therefore in charge of creating appropriate views on the data models according to the different stakeholders, materialising them when needed. At the same time the dashboard will offer functionalities of setting parameters on the game.

3.4.2.3 Marketplace

The marketplace component has multiple uses:

- Primarily it is the place where Budget Holders can get access grant for their schools/institutions to games, also offering reports on usage and implementing the chosen business models, including accounting/billing where appropriate.
- It also provides an interface for browsing available games by teachers and carers.
- It is also the place from which it is possible to activate the installation procedure for an available game. The complexity of installing a game varies a lot according to the software and hardware required on the client side. Clear instructions and where appropriate support will be provided through the marketplace.
- Beside and beyond the installation phase, the marketplace works together with the dashboard for providing the teachers/carers an easy way to start games.

3.4.2.4 Logging Component

The logging component is responsible for storing all the data relevant for the prosociality in the right dataset. In future releases where the current constraint of having a unified log format for the game will be relaxed, it will also host the transcoding module able to ingest game log interpreting them according to the PSL model.

3.4.2.5 Game deploy and governance manager

The PSL platform will support an efficient gaming applications deployment and governance function in a virtualized environment in a PaaS fashion. PSL itself is meant to be a PaaS infrastructure enabling the deployment and governance at runtime (operation) of gaming applications on a centrally-managed platform.

The governance manager functionality provided by the PSL platform receives a Deployable Game Model (DGM) through its Deployer API and later on, will execute the DGM on a virtualized environment. The DGM contains the plan describing the steps to deploy the Game application as well as the artefact representing the software. The Deployer API may be cloud-agnostic; it should be able to deploy gaming applications on different virtualized environments using third party adapters.

Once the game has been deployed, the PLS platform will offer mechanisms to fully govern the lifecycle of the software (Start, Stop, Replicate, Un-deploy).

Finally the PLS platform will provide a performance monitoring functionality based on unified “cloud offering” independent metrics, such as latency/response time, bandwidth, application status, etc., to allow gaming operators to proactively monitor the health and the performance of gaming applications.

3.4.2.6 Adaptation engine

A core component of the platform is the adaptation engine. This engine utilizes an optimization mechanism that relates multiple indicators of the players state (prosociality level, engagement, and ability) to the adjustments that a game needs to make in order to increase those measures.

ProsocialLearn imports new constraints in game adaptation research since the mechanism needs to be independent from any prosocial game. There are a number of optimization methodologies that fit the needs of such a generic adaptation mechanism. In an online learning scheme an algorithm tries to gradually learn the prosocial state to game mechanics parameters relation from examples collected during gameplay. Stochastic search is another method that can be used to discover the right set of game parameter values (solutions) that maximize the prosociality of a given player without any prior knowledge about the player or the game parameters.

The current state-of-the-art of research in adaptation and personalization can provide a basis for the construction of the mechanism. Yannakakis[19] developed an architecture for online game adaptation by training a neural network as the player’s cognitive model and then adapting a game towards the maximum level of player engagement through the use of gradient ascent optimization. K. Stanley’s *rtNEAT* algorithm which was used in the Galactic Arms Race game[20], implicitly tracks user statistics and tries to personalize the game via real-time neuro-evolution. Also, in their review of game design[21], the authors used an interactive evolutionary framework that explicitly queried the user in order to discover the user’s preference model.

The choice of the right methodology to be adopted strongly depends on the nature of the search space but also on the time complexity demands that online adaptation poses.

3.4.2.7 Observations acquisition manager

ProsocialLearn enables multichannel acquisition of observations. Each channel has its own architecture that is related to the nature of the observation. Currently, a wide range of vision and audio signals are supported by the platform.

Facial Expression Analysis

Facial expressions are generally considered a strong indicator of a person’s emotional state. As a form of nonverbal communication, expressions need to be recognized using vision-based automated frameworks for tracking and analysing activity of muscles and muscle groups, as defined by Ekman’s Facial Action Coding System (FACS)[22]. The ProsocialLearn observation platform will utilize a single, High-Definition (HD) webcam mounted on top of the player’s monitor, in order to properly extract and analyse user facial activity in real time. For processing each consecutive camera frame, two main

[19] Yannakakis, Georgios N., and John Hallam. "Real-time game adaptation for optimizing player satisfaction." *Computational Intelligence and AI in Games, IEEE Transactions on* 1.2 (2009): 121-133

[20] Hastings, Erin J., Ratan K. Guha, and Kenneth O. Stanley. "Evolving content in the galactic arms race video game." *Computational Intelligence and Games, 2009. CIG 2009. IEEE.*

[21] Liapis, Antonios, Georgios N. Yannakakis, and Julian Togelius. "Limitations of choice-based interactive evolution for game level design." *Proceedings of AIIDE Workshop on Human Computation in Digital Entertainment*. 2012.

[22] Ekman, Paul, and Wallace V. Friesen. "Facial action coding system." (1977).

components will be used to exploit the facial information, namely a face detector (based on the Viola-Jones object detection algorithm[23] implemented in OpenCV[24]) and a facial feature extraction algorithm, as depicted in Figure 18. In cases where face detection is difficult, extraction of the face image colour histogram can be utilized.

Under FACS, facial expressions are described by Action Units (AUs). Subsequently by combining AUs emotional states can be described. In order to define these AUs and extract the information from the changes of their state, an Active Shape Model (ASM) will be utilized. ASMs are statistical models of the shape of faces which iteratively deform to fit to an input camera frame[25]. To ensure that the framework will be able to identify displacements of facial features even at a micro-expression level the ASM will employ a large number of features. Tracking the movement of these facial feature points will enable the expression analysis module to identify the activation of specific AUs and thus produce a measure of the player's real time emotional state.

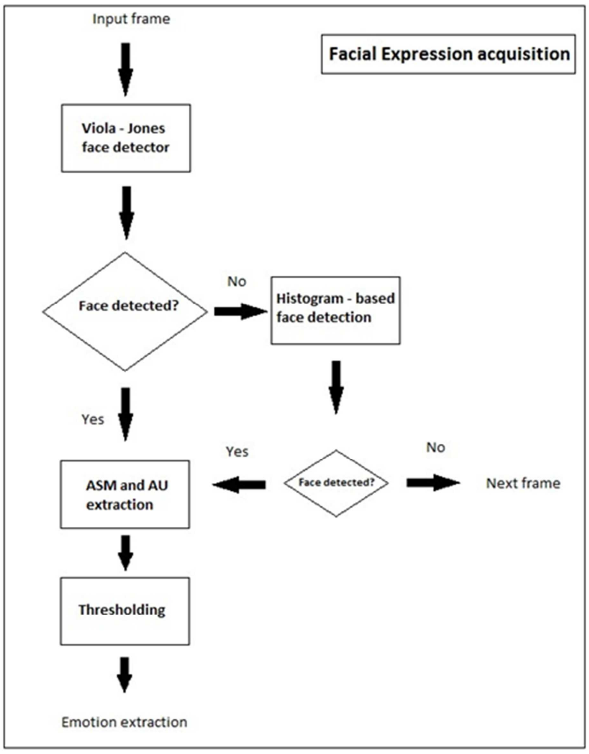

Figure 18 - Overview of the facial expression acquisition component.

Eye-Gaze Tracking

Eye-gaze as a form of human-computer interface holds great promise for studying players' interaction with the game environment. Eye gaze data has been reported to provide information on

---

[23] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proc. of the 2001 IEEE Computer Society Conference on* (Vol. 1, pp. I-511). IEEE.

[24] http://opencv.org/

[25] Cootes, T. F., Taylor, C. J., Cooper, D. H., & Graham, J. (1995). Active shape models-their training and application. *Computer vision and image understanding*, *61*(1), 38-59.

emotional state[26], and may as well serve as an estimate for player engagement[27]. To validate these ideas an eye-gaze tracker will be employed to operate alongside the facial expression. Calibrated as well as un-calibrated gaze tracking frameworks will be explored for unobtrusive integration with the gaming platform. Figure 19 presents an overview of the system. The component will generate output on features covering eye gaze, such as gaze location on the screen, gaze distance, blinking and pupil diameter.

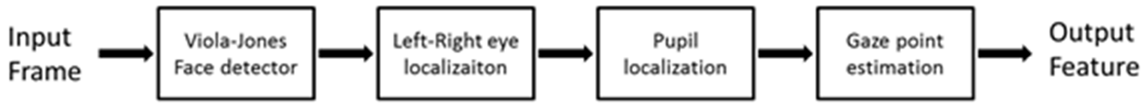

Figure 19 - Overview of eye-gaze point estimation

Voice analysis

The acquisition of voice consists of building a system to take a sampled time varying signal and represent it in terms a number of useful features. The specific features employed are based on evidence from the literature[28] as to features which are shown to provide efficacy in emotion detection. An overview of the acquisition process is shown in Figure 20.

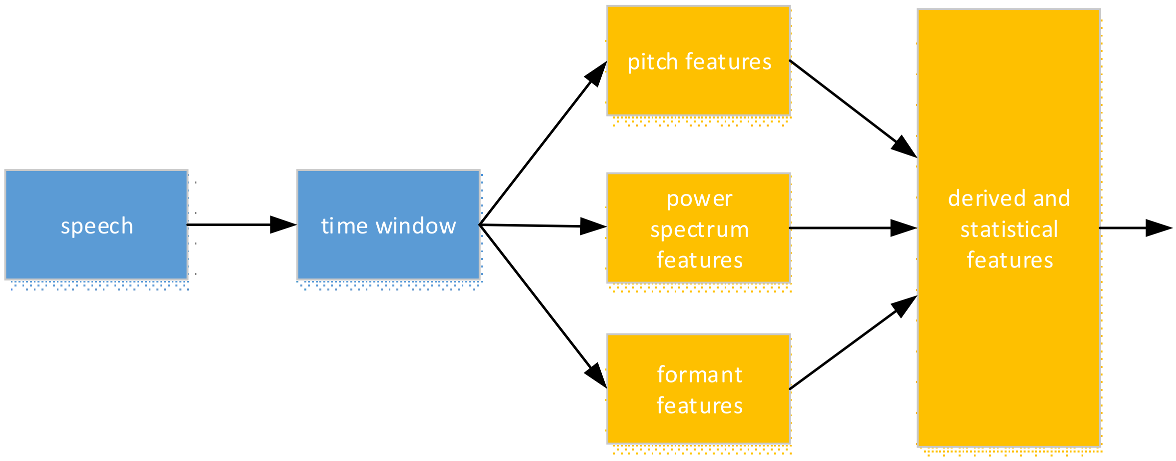

Figure 20 - Overview of the voice acquisition component

The incoming speech signal is captured either in game or parallel to the game. This signal is windowed and features are computed within a time window. A window is needed in order to perform frequency domain analysis. The features extracted are in one of three classes: pitch, power spectrum, and formants (distinctive frequencies in signal). The initial extraction is used to derive additional features. Using pitch as an example the pitch feature can show the track of the fundamental frequency over time. In addition it is possible to compute the mean, variance, range, and first order differences. All features collected this way are augmented into a single feature vector which is passed onward to the fusion system. The dimensionality of this feature vector depends on the number of features extracted.

---

[26] Soleymani, M., Lichtenauer, J., Pun, T., & Pantic, M. (2012). A multimodal database for affect recognition and implicit tagging. *Affective Computing, IEEE Transactions on*, *3*(1), 42-55.

[27] Nakano, Y. I., & Ishii, R. (2010, February). Estimating user's engagement from eye-gaze behaviours in human-agent conversations. In *Proc. of the 15th Int. Conf. on Intelligent user interfaces* (pp. 139-148). ACM.

[28] F. Dellaert, T. Polzin, and A. Waibel. Recognizing emotion in speech. In Proceedings of fourth international conference on spoken language processing, volume 3, pages 1970-1973, 1996.

Multimodal Fusing component

The purpose of the fusion component is to combine the variety of input modalities and assert a best estimate for emotion, engagement, and also possibly player ability that, as stated in the description of work, will be investigated together with emotion and engagement as one of the source of the adaptation algorithms. These independent estimates will be used to provide the inputs necessary for the adaptation component and user profile. A simplified view of the architecture is shown in Figure 21 for single output estimate.

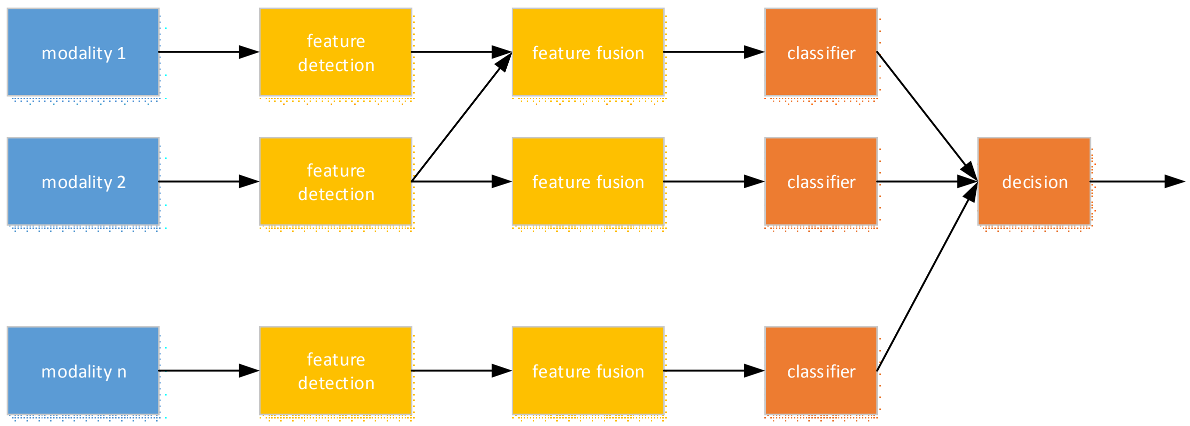

Figure 21 - High level view of fusion component

The input to the component is one of a number of input modalities. For each specific modality features are extracted. Features are then combined and input into a classifier which provides an output. Individual classifiers will be used for each of emotion, engagement, and player ability. The classifiers are supervised and pre-trained on an input set of labelled features. The individual classifier outputs are then pooled to make a final decision based upon the input features. In this view the fusion happens at two specific levels. Firstly, there is feature level fusion and secondly there is decision level fusion.

Feature level fusion takes place within the classifiers. In our situation there are a number of different feature detectors where the detectors generate features in a high dimensional space. When the feature vectors are combined from different modalities this problem is compounded. Building classifiers in this scenario is problematic and they are prone to high misclassification rates. In order to ameliorate this problem a number of approaches can be employed:

Dimensionality reduction: techniques to embed a high dimensional space into a lower one. Normally, implicitly assumes that the data lies on a manifold in the high dimensional space. Preferred approaches here will be non-linear dimensionality reduction techniques such as Laplacian Eigenmaps or locally linear embedding.

Feature selection: techniques that employ search and an evaluation metric to evaluate the feature set. These techniques build an alternate representation of the data. Preferred approaches here will be clustering, support vector machines, and naïve Bayes.

In reality it is proposed that we use a combination of these where appropriate to the underlying features. Fusion at this level will allow the exploitation of dependencies within features. These algorithms are understandably complex and effort will be made to reduce the computational overhead.

Decision level fusion is performed based upon the outputs from the individual classifiers. The benefit of this second fusion step is primarily to provide robustness to missing observations. Furthermore, the combination allows the possibility of estimates of confidence based on the pooled output of the individual classifiers. The general approaches we can use here are:

Aggregate methods: techniques which pool the various outputs in a simple or naïve manner. These can be as simple as taking the most common response or use a simple metric such as most common.

Statistical methods: techniques that use statistical frameworks to pool the outputs. These include methods based on Bayes or Dempster-Shafer.

Information Theoretical methods: techniques that estimate the informational content and employ this so as to pool the outputs. Favoured techniques of this type include mutual information or Kullback-Leibler divergence.

The favoured approaches here are either aggregate or statistical methods. In contrast with the feature level fusion this method of fusion is comparatively simple and has minimal computational overhead.

Depending on the type of fusion the operations can be performed either on the client or a centralised server. Feature level fusion can operate easily on a client. In fact in terms of reducing the bandwidth of transmitted data this makes sense. In the case where the inputs from multiple modalities are fused then the fusion will operate upstream from the capture device. Finally, the final decision level fusion operates at the server as it requires the outputs of all the individual classifiers.

3.4.2.8 Ontology manager

The ontology defines the shared knowledge definition across the users of the platform and the platform itself. It should therefore be queryable both through a graphical interface and through automatic machine interaction. The graphical interface should be made available both in the dashboard for supporting game choice by the teachers and in the marketplace for supporting game developers and budget holders.

3.4.2.9 User model manager

The large amount and the importance of user related information stored in the platform requires a dedicated module responsible for managing it. In particular:

- it implements the process for binding the anonymous data with the identity information;
- it supports the different possible views on the user model related to the different stakeholders; and
- it offers a service interface through which different modules can access the user data.

3.4.2.10 PSL state manager

As seen in Section 3.4.1.6 the prosocial measurable space is very complex and related to a context. This module is able to virtually create and manage the proper prosocial measurable space identified by the context and the core domain to which the data are bound.

Each space has for each of its dimensions metrics defined as functions over data coming from a source (external observation, in game data or user profile). The module applies the proper functions to the incoming data, so that it is possible to assess the prosocial state of a user. Fast and reliable access to all the necessary data is required by this module for the effective and timely execution of the analysis.

3.4.3 Data flows

This section summarizes the data flows in the different use cases.

3.4.3.1 Secure Access data flow

All the users, with independence of the entry point in the system, must be identified. This is valid for game players, teachers, budget holders or developers, but the system of identification may be different, as for children we want a simplified system. In any case, sensitive data will be stored separately, and will only be accessible by the Secure Access module. Other modules will work with anonymized identities.

Another important function is role management. Each user will have a role in the system, and other modules will ask the Secure Access component about the role and capacities of the user before executing certain operations.

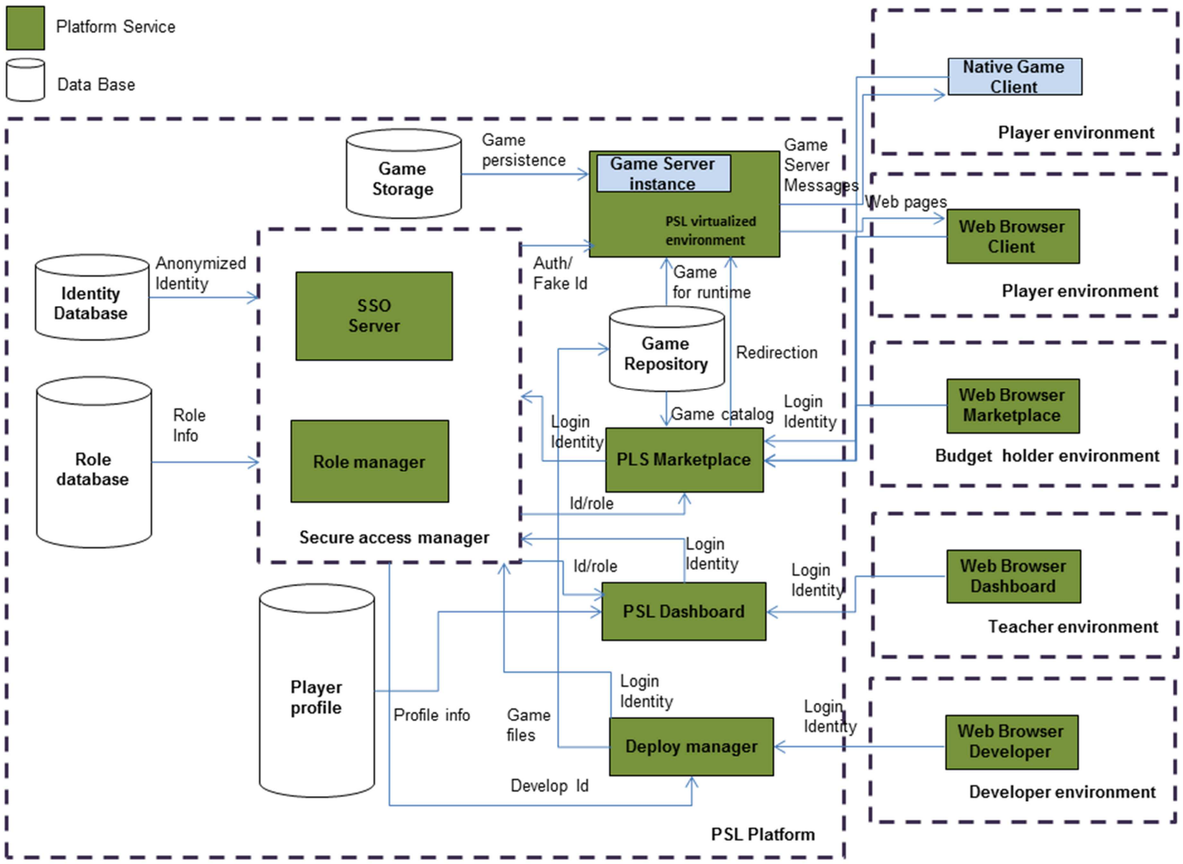

Figure 22 - Secure Access data flow

3.4.3.2 In game data flows

Figure 23 shows the data flow in the in game scenario. It is characterized by multiple interaction modalities with the users (I/O devices and sensors). It exploits general services like the external synchronization and the Single Sign On server. The description of the game logic with the related components is game specific and therefore out of the scope of this deliverable. Instead it is worth highlighting that emotion extraction is partially performed on the client side and that sensitive data not authorized to leave the platform are erased locally in accordance with ethical and legal prescriptions. The diagram clearly shows that two independent data flows come out from the client and that they are later jointly used in the platform environment to assess the current prosocial state of the player.

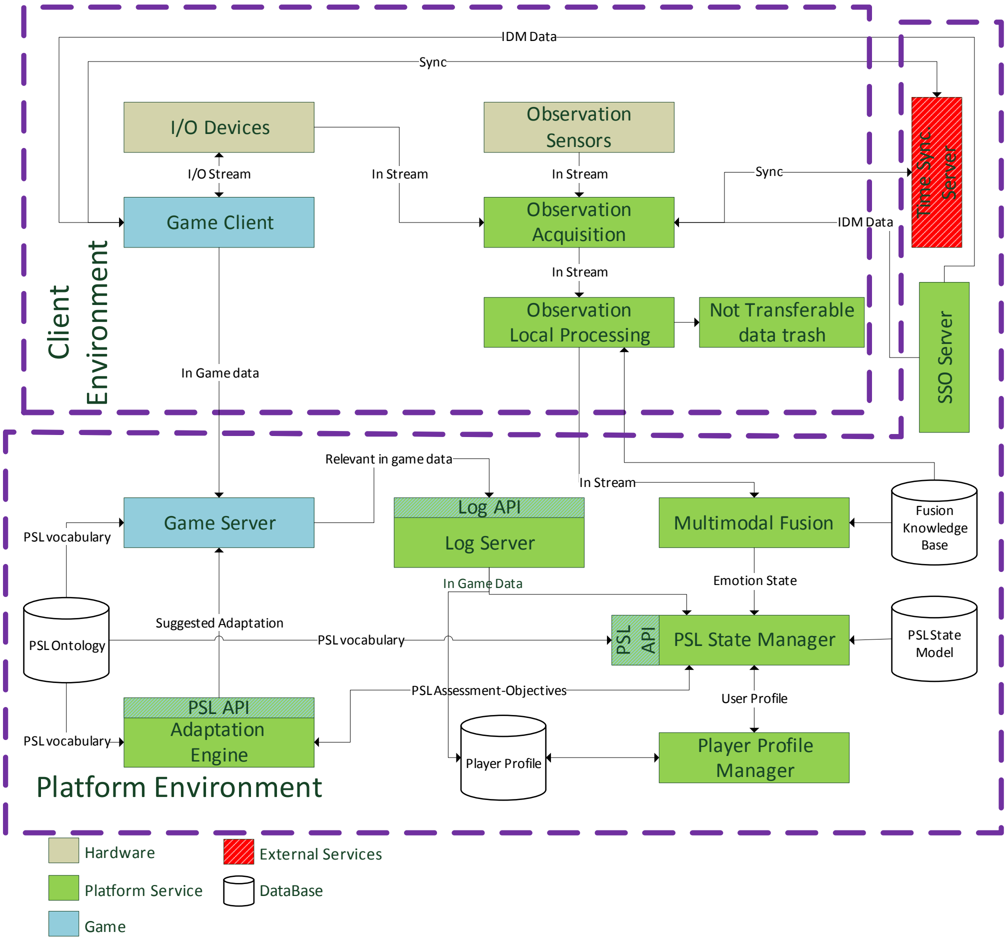

Figure 23 - In game data flow

3.4.3.3 Dashboard data flows

Figure 24 shows the data flows across the different components involved when the end-users use the dashboard.

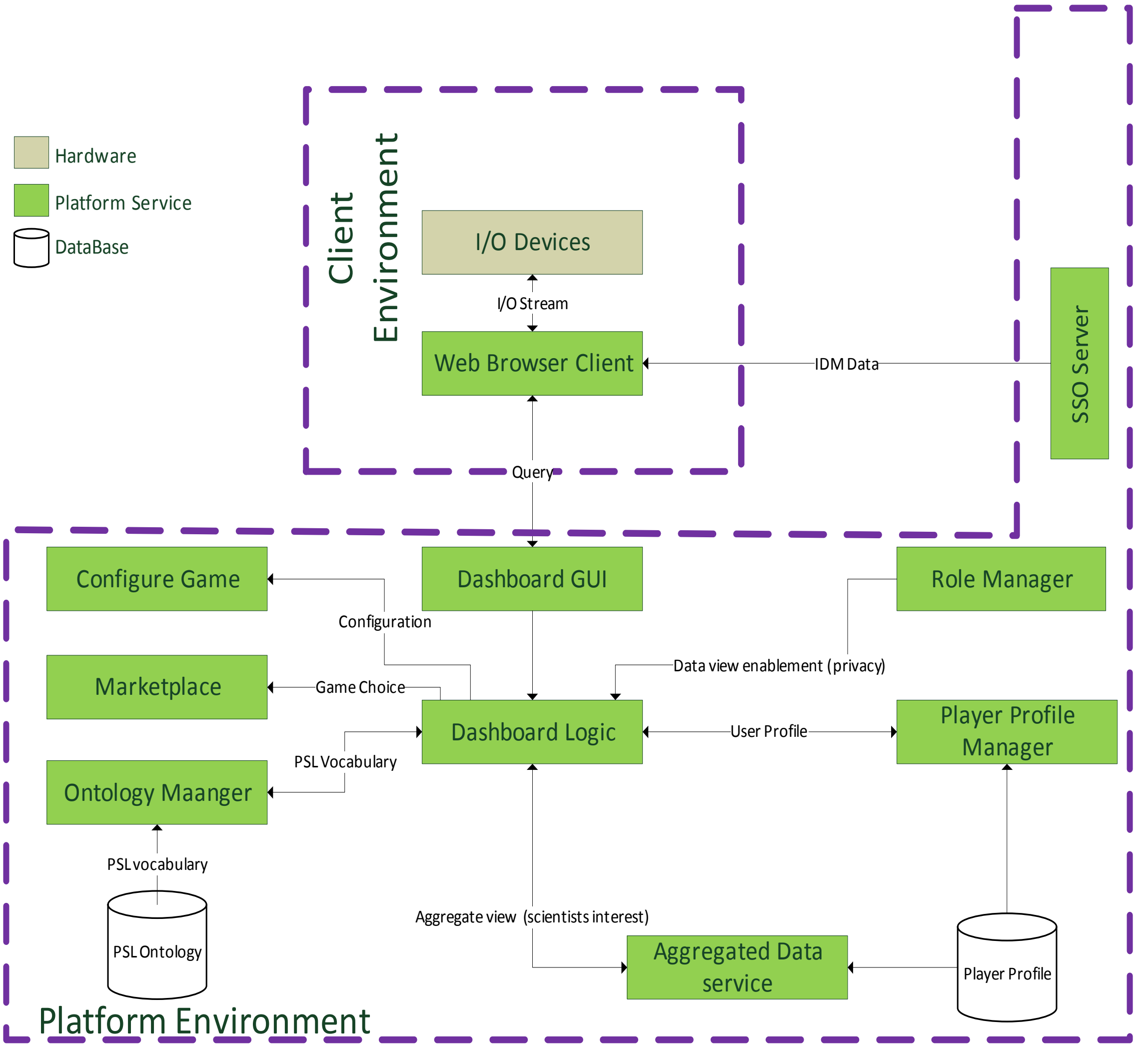

Figure 24 - Dashboard data flow

3.4.3.4 Game deployment data flows

The server part of the games will be deployed on a PaaS platform in a cloud environment. The client part will be executed in the browser, in the case of web applications, or in the case of native games, it will be installed and executed in the client device, communicating with the server backend in any case.

3.4.3.5 Marketplace data flows

The marketplace is the entry point for players, especially in the case of web games. The player can see a list of games that he/she is allowed to use and choose one.

For the budget holder, the complete list will be show. According to the business model, they will be able to pay for games and make it available for some players.

The game developers do not use the Marketplace, but using their Deployment Manager they provide the games and the information (text, screenshots) used in the Marketplace.

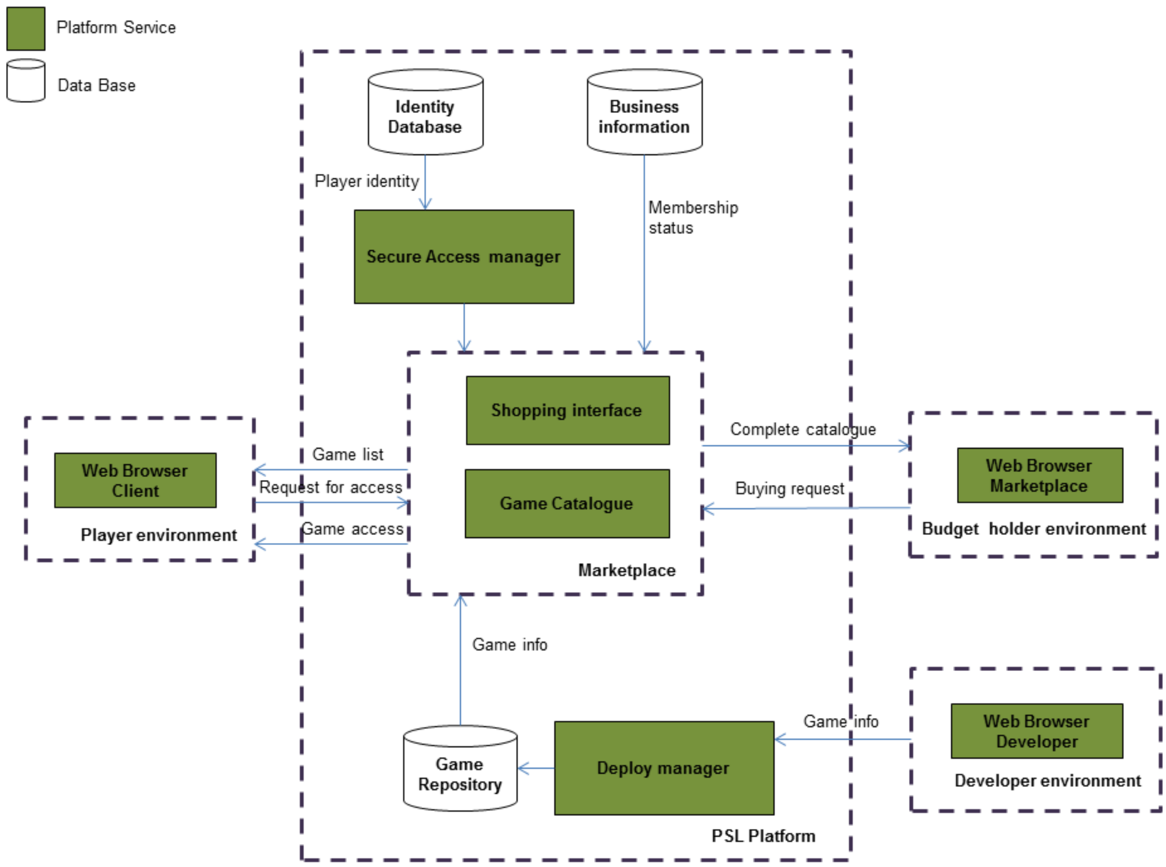

Figure 25 - Marketplace data flow

3.5 Technology architecture

In TOGAF terms the technology architecture describes the binding between abstract data and software modules described previously and technical choices. The technological solution to implement the above mentioned modules has not yet been finalised and this section refers mainly to the solutions adopted for implementing the cloud.

ProsocialLearn will rely on an open cloud architecture. At a high level, this means that the cloud architecture implemented by ProsocialLearn will be based on a solid composition of open source components and solution and where possible, implements open standard formats, APIs, and protocols that deliver both interoperability and value to teachers/local public administration, game providers and software operators. Figure 26 below highlights the open PSL architecture.

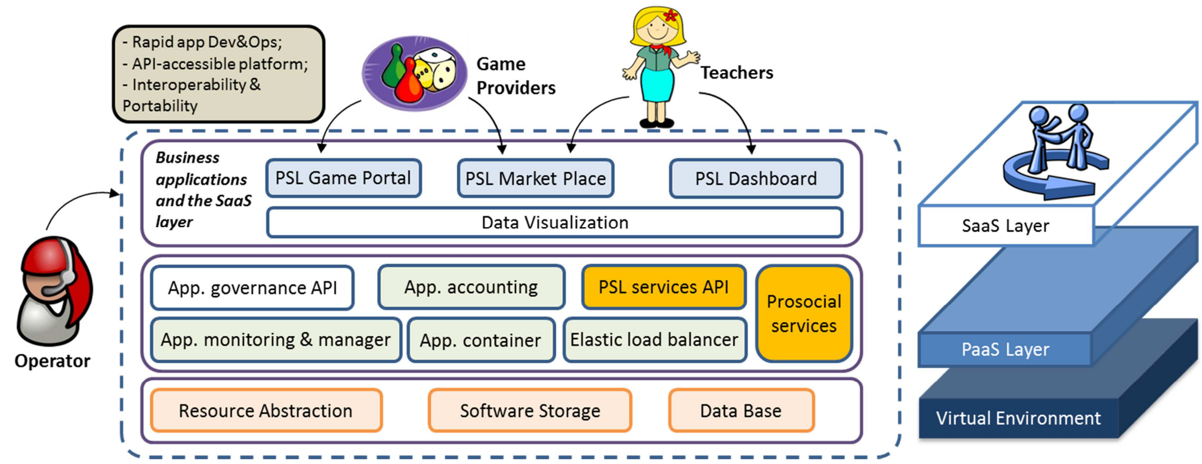

Figure 26 - Cloud architecture

Our architecture extends the cloud stacks, PaaS and SaaS, while it relies on a robust IaaS.

At the SaaS layer, our open PSL architecture addresses business, user interface and data visualization issues.

Continuing up the technology architecture stack, PSL is still exploring a number of open alternatives for building up a robust PaaS layer. However, during the editing of this deliverable, the ProsocialLearn consortium is already beginning to investigate the Cloud Foundry community (almost a de facto standard for Platform as a Service environments). PSL is planning to integrate the Cloud Foundry technology into the added value services of PSL. Cloud Foundry provides a platform that allows game providers and game operators to deploy, manage and scale cloud-based games quickly, and easily.

The App. container supports a wide variety of runtime environments (including Java, JavaScript, Python, Ruby, etc.), popular frameworks (such as Node.js, Tomcat, and Ruby on Rails, etc.), and a rich variety of add-ons services (NoSQL databases, queuing, messaging, caching, analytics, social media, big data, etc.).

On the other hand, it provides an open architecture for the incorporation of additional runtime environments, frameworks, and services through the use of the buildpack technology. Finally, Cloud Foundry PaaS also provides an open approach to integrating with the underlying infrastructure. This allows Cloud Foundry to be run on Amazon, VMware, and most importantly with IBM's open cloud architecture, OpenStack.

- PSL Services API: The PSL services APIs will provide an easy mechanism for game developers to integrate into their games all the prosocial functionalities implemented by the PSL platform. The APIs need to be self-explanatory, well documented and language programming agnostic.
- PSL Game delivery models: While PSL relies on Cloud Foundry functionalities which are designed to support multiple programming languages it will support multiple game delivery models as well.
    - PSL Online games: games are fully operational remotely (in the cloud) through the Internet. Games are played through the common browser. Game engine/servers and the PSL prosocial services are fully governed by the PSL platform. The advantage of this models is that, easily the game is platform/Operating System as well as device agnostic;

any devices (tablets, mobiles, PC) any platform and Operating System: Windows (Microsoft), Android (Google), Mac iOS (Apple)
    - PSL Fully downloadable games: by contrast to the previous model, here, the game, before being played needs to be fully downloaded onto the player device. In this case, the game is not agnostic to the device. PSL needs to support the delivery of multiplatform games. Games “consume” the PSL prosocial functionalities interacting with the PSL services, run in the cloud.
    - Hybrid model. Games are partially downloaded at independent market place or official app stores (when the device is a mobile or tablet). Also here, the game is not agnostic to the device. PSL needs to support the delivery of multiplatform games. Games “consume” the PSL prosocial functionalities interacting with the PSL services, run in the cloud.
- App monitoring and manager: A range of standardized and unified games metrics of different types (low level, container level, app level, etc) allows runtime monitoring of running games so as to assure a reasonable end-to-end Quality of Service (QoS).

The ProsocialLearn PaaS and SaaS layers will rely on a robust and scalable virtual environment providing the foundation for the PSL operation.

3.5.1 Support for platform deployment

The platform needs to accommodate different deploy modalities according to different scenario. A set of minimal requirements that the school environment needs to support is under development. In any case flexibility must be supported, especially during the project when several trials are going to happen. Currently the standard solution is to have a client environment connected to a server environment located elsewhere and accessed through internet via a standard broadband connection. Completely isolated solution where all the client and server sits on a private LAN created ad-hoc in the physical environment of the client (e.g. a school) is also envisaged. Hybrid solutions are also possible. The process for specifying the trials procedure will include the deployment solution chosen for the given situation.

4 Conclusions

This deliverable has presented the first version of the ProsocialLearn architecture.

It produces a globally coherent view across the research activities in the different work packages also providing a formal base to be used by the implementation team to develop the different architecture modules.

Because of the relatively early phase of the project (M6), within the general goal of designing the system's architecture, the accent in this document has been on the requirements and on data modelling. The next iteration of the document (due at M15) will complement and revise the current version providing all the necessary clarifications.

Appendix 1 – Example of Game-specific run-time model

Developers of prosocial games will need to map their 'concrete' game concepts to the PSL core ontology by carefully selecting appropriate elements from their game design and encoding them as derivative types. In the model based example below (Figure 27), we show a fragment of the PSL short study game 'Labyrinth':

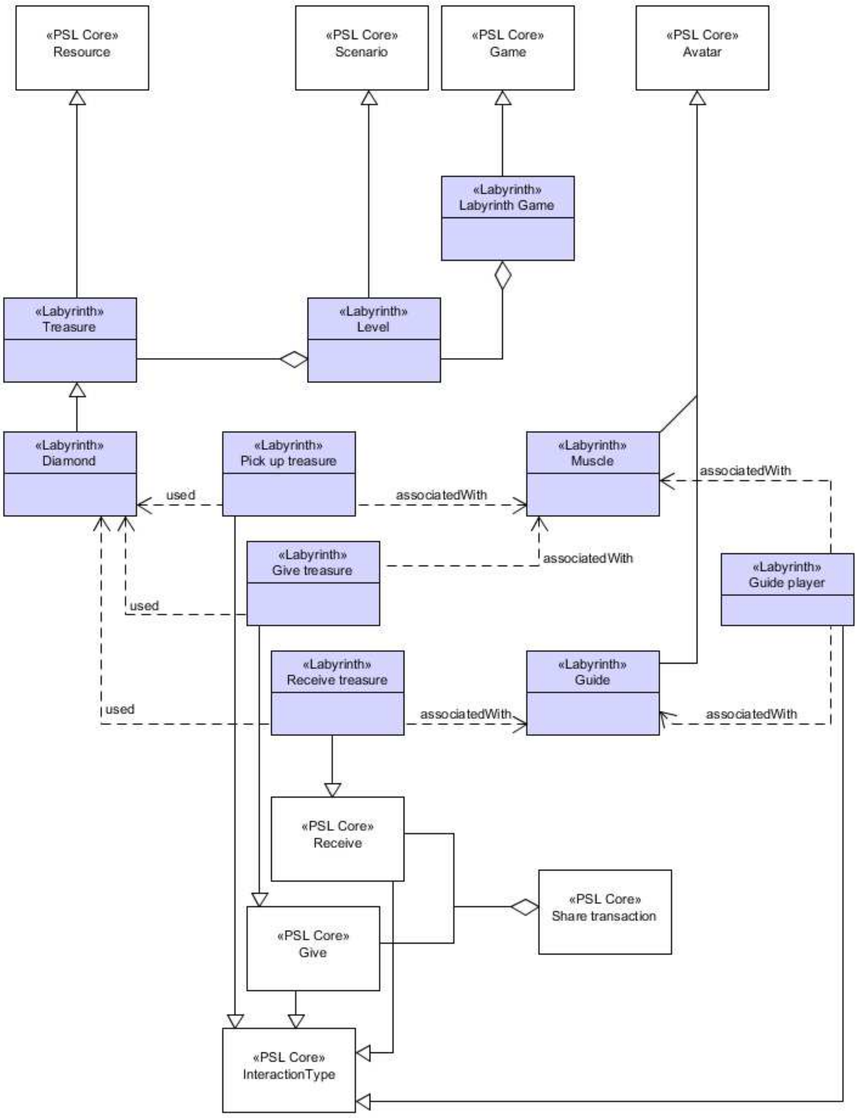

Figure 27 - example of game modelling

Here we see a sub-set of Labyrinth game types[29] (depicted in blue) mapped to PSL core ontology types; for the sake of brevity, details relating specific attributes and relations between the Labyrinth

[29] These types have been specified only for the purpose of illustration in this document and do not reflect the actual implementation of the Labyrinth game *per se*.

classes are omitted. Concrete elements of the Labyrinth design such as the game instance, its levels, ‘treasures’ and avatars (the ‘muscle’ and ‘guide’) are mapped to the structural elements of the core ontology at the top of the figure. Labyrinth specific interactions (*pick up treasure*; *give* and *receive treasure*) are mapped to higher level ontology interaction types. Two of these interactions, *give* and *receive*, match to represent a *share transaction* (in its simplest form). To understand the transaction more fully, we use the W3C provenance based predicates to link activities to entities and agents. For example, we can link the act of giving a diamond by the *muscle* to the *guide* by tracing a graph of W3C provenance predicates (*associatedWith* and *used*). Starting from act of giving (*Give treasure*), to the diamond instance (*Diamond*), to the act of receiving (*Receive treasure*).

Formally, this run-time data model will be expressed as a derived ontology that will be used by the game adaptation engine to interpret the specific activities of the Labyrinth game. To assist developers in their implementation of PSL concepts in code, a user definable ontology look-up-table (LUT) and API will be provided make recording PSL based game data simple. It is anticipated that this run-time data will be acquired by the PSL platform using one of two methods: i) *bespoke prototype log parsing* or ii) *direct model based observation logging*. In the former case, expected to be used in the project’s early studies for validation purposes, a bespoke logging format will first be generated by the game prototype software. This will then be transformed off-line into the model based observation data structures described in this document and ingested by PSL services for experimentation and testing purposes.

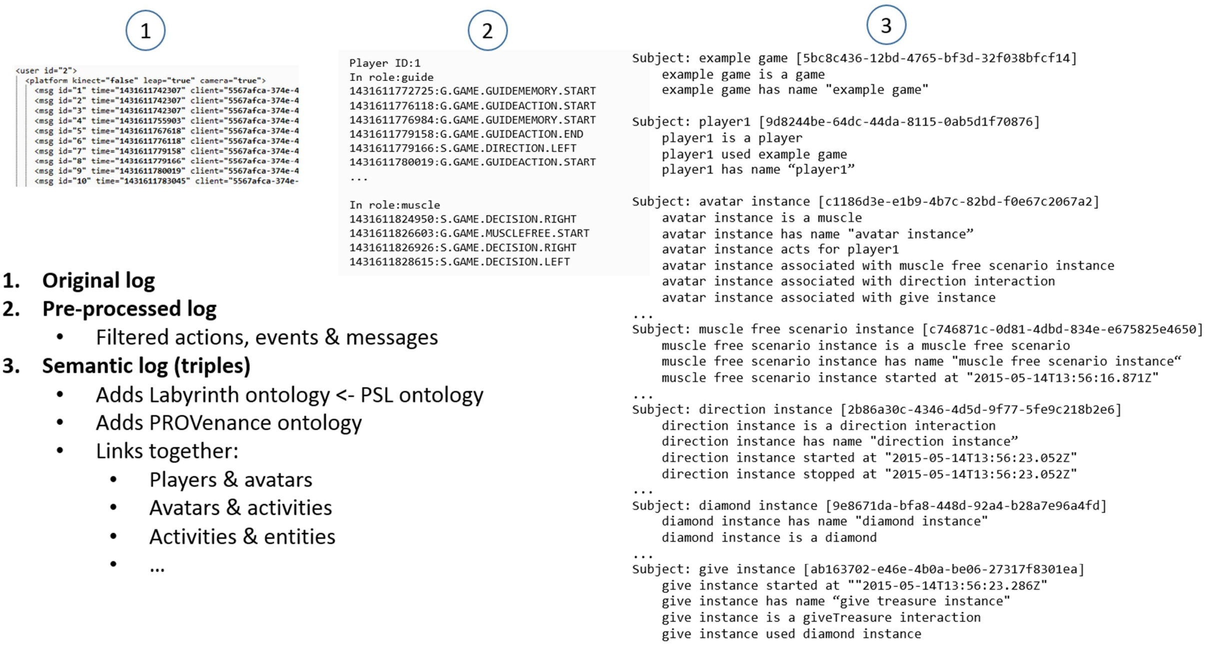

Figure 28 - 3 phase transformation process for bespoke prototype log parsing

In Figure 28 we illustrate this process using a fragment from an early version of the Labyrinth game logs in which game events are transformed into a set of semantic, model based records of game interactions.

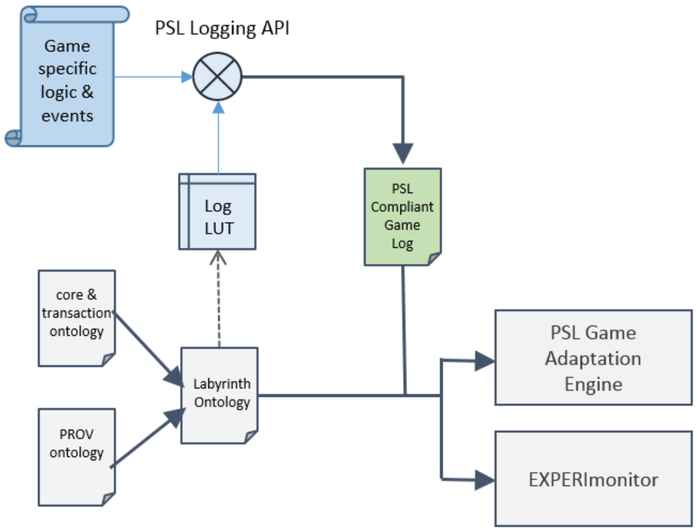

Figure 29 - Direct model based observation logging

Following on from this initial processing of game logging, game developers will be provided a model based API that will allow them to directly represent their game logic as PSL constructs based on the core PSL ontology. In Figure 29we illustrate this process for the Labyrinth game example.