Infrastructure for Secure Sharing Between Picture Archiving and Communication System and Image enabled Electronic Health Records

Krupa Anna Kuriakose
MASc Candidate
Dept. Electrical, Computer and Software Engineering
UOIT
Krupa.Kuriakose@uoit.ca

Supervisor : Dr. Kamran Sartipi

February 22, 2013

Overview

- Drawbacks of the existing PACS
- Proposed solution
- Introduction to OpenID and OAuth
- Case Study : E-health Services with Secure Mobile Agent

Current security issues in PACS

Lack the following features :

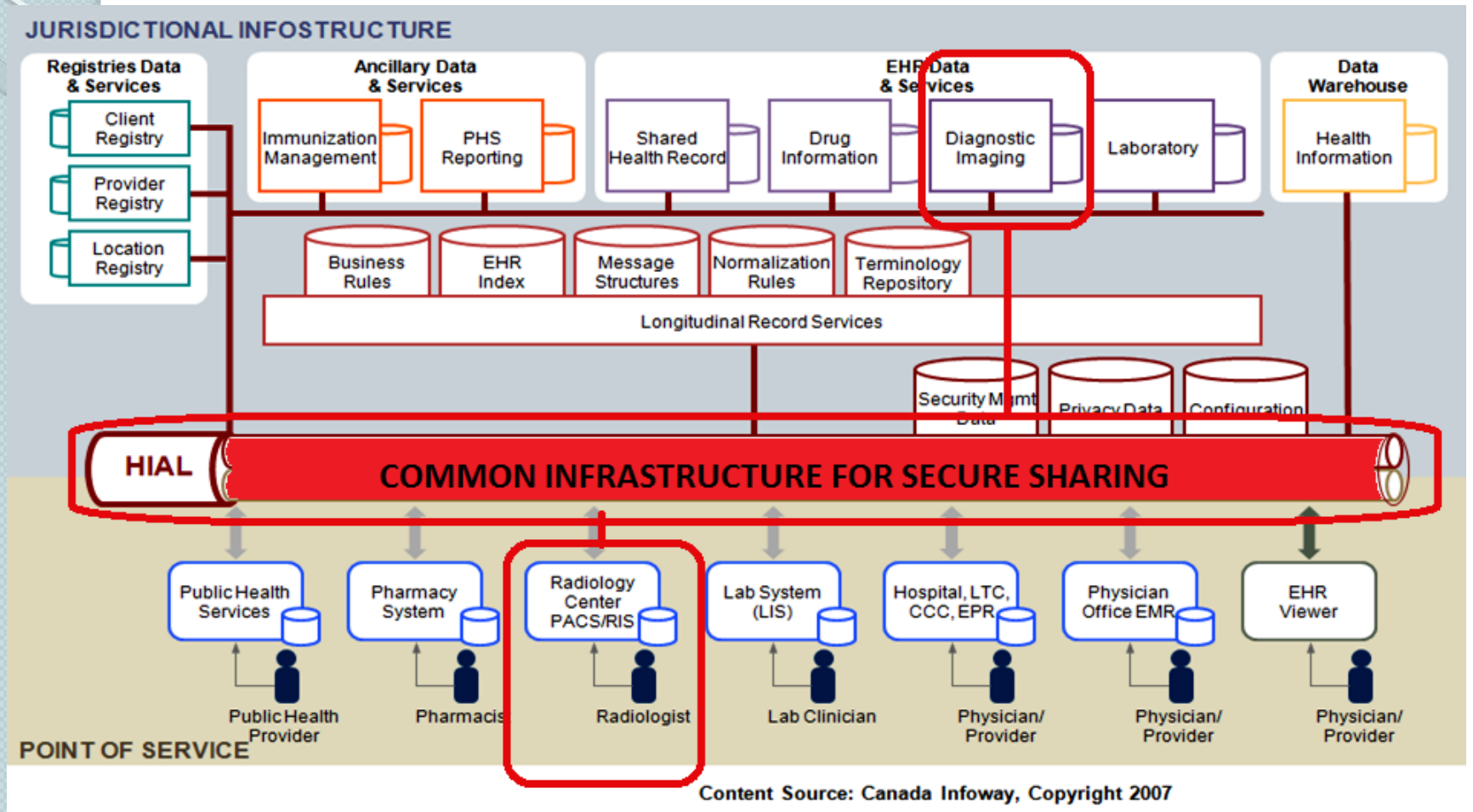
- Infrastructure for Federated Identity Management (FIM)
- Common set of access control policies
- Integration of patient consent directives with the security policies
- User authentication and audit to data is local to each system and not federated

“ PACS have no means to integrate and interoperate with common infrastructure”

Solution to address the issue

- A token based User Registry to initially authenticate users
- A Consent Registry that holds the consent directives defined by patients
- A Health Information Access Layer with a standard messaging and communication protocol

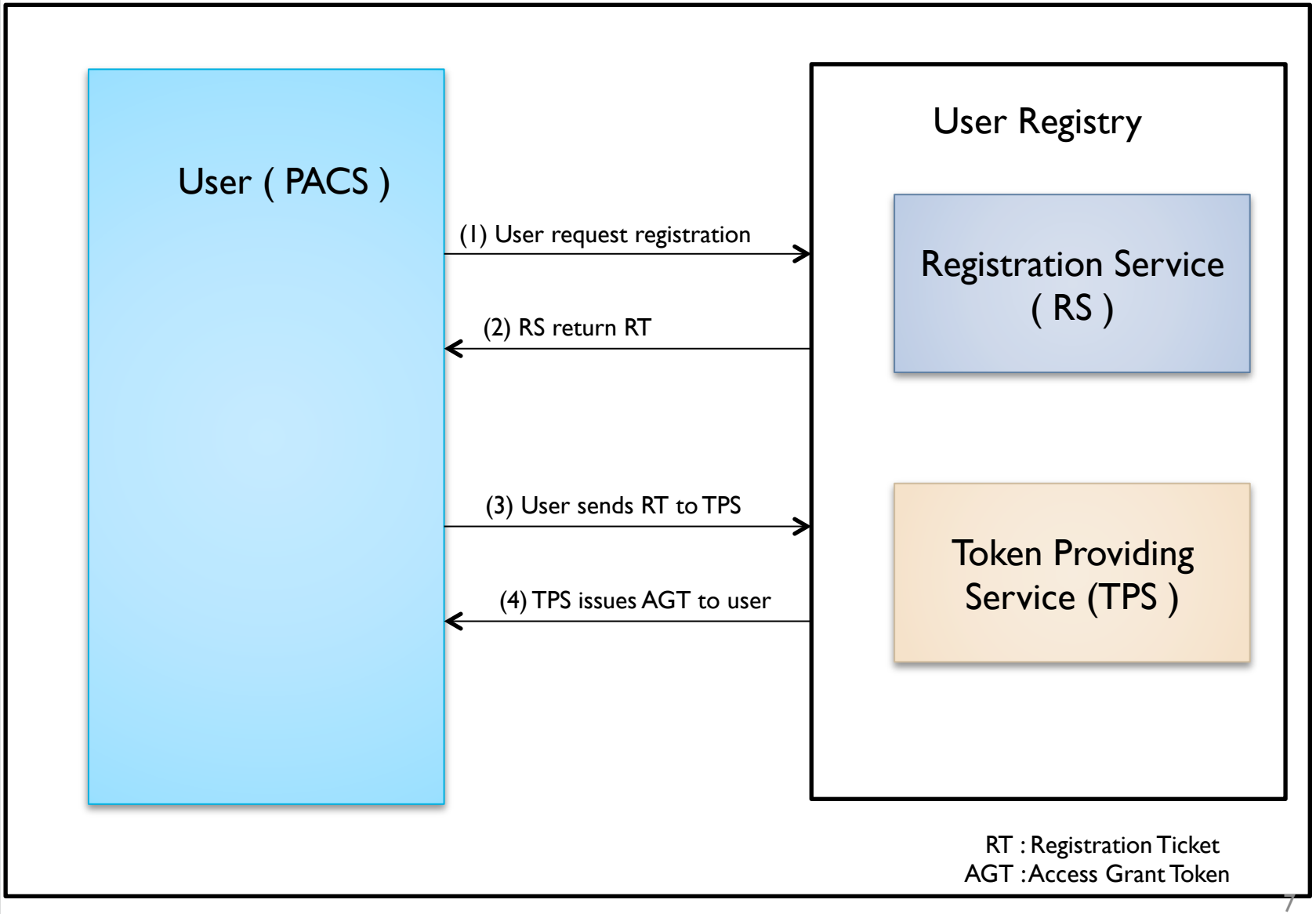
Research Area



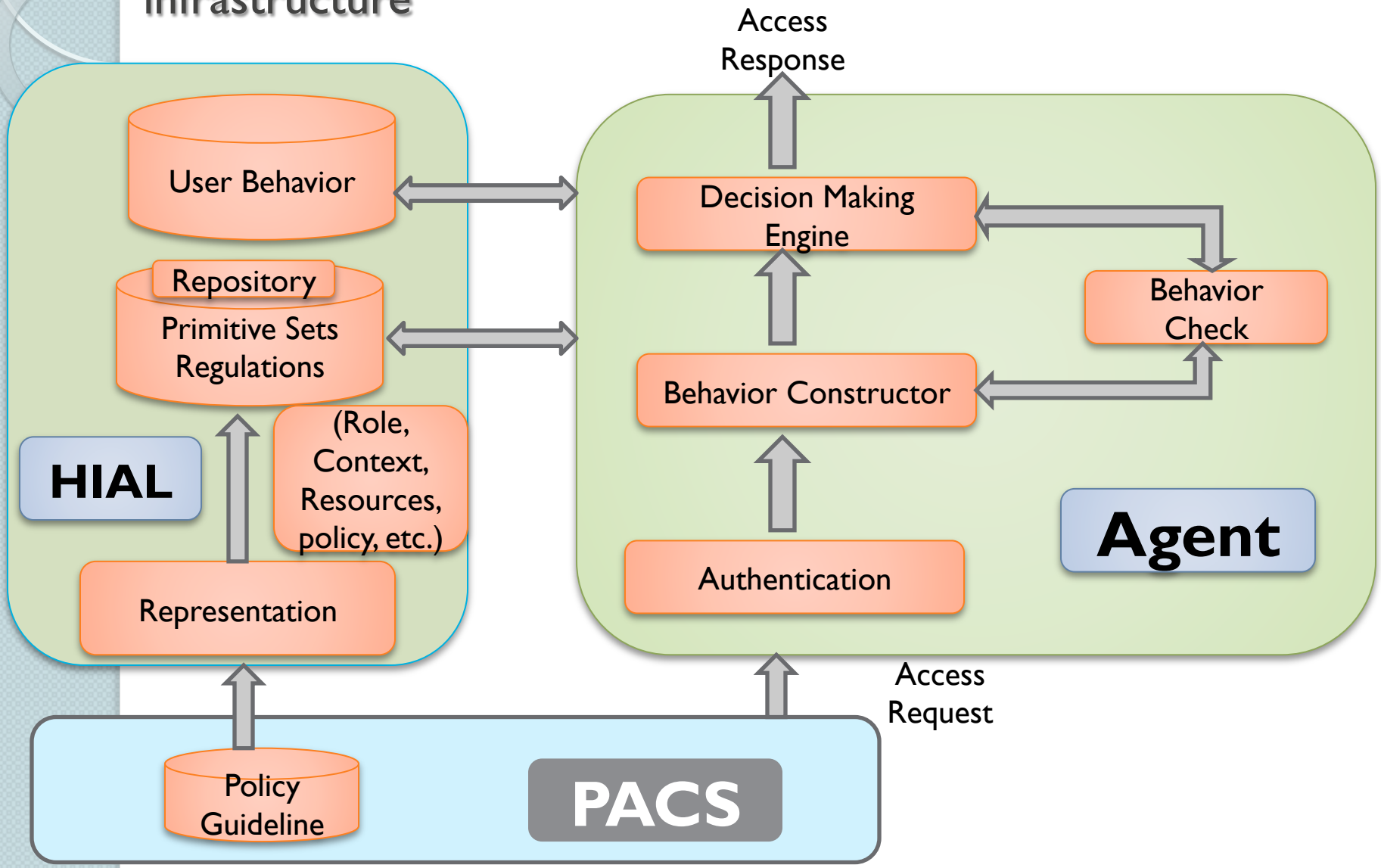
Proposed Solution

- Stage 1 : Token based authentication of the user prior to sending access request to EHR
- Stage 2 : Agent managed behaviour based access control infrastructure

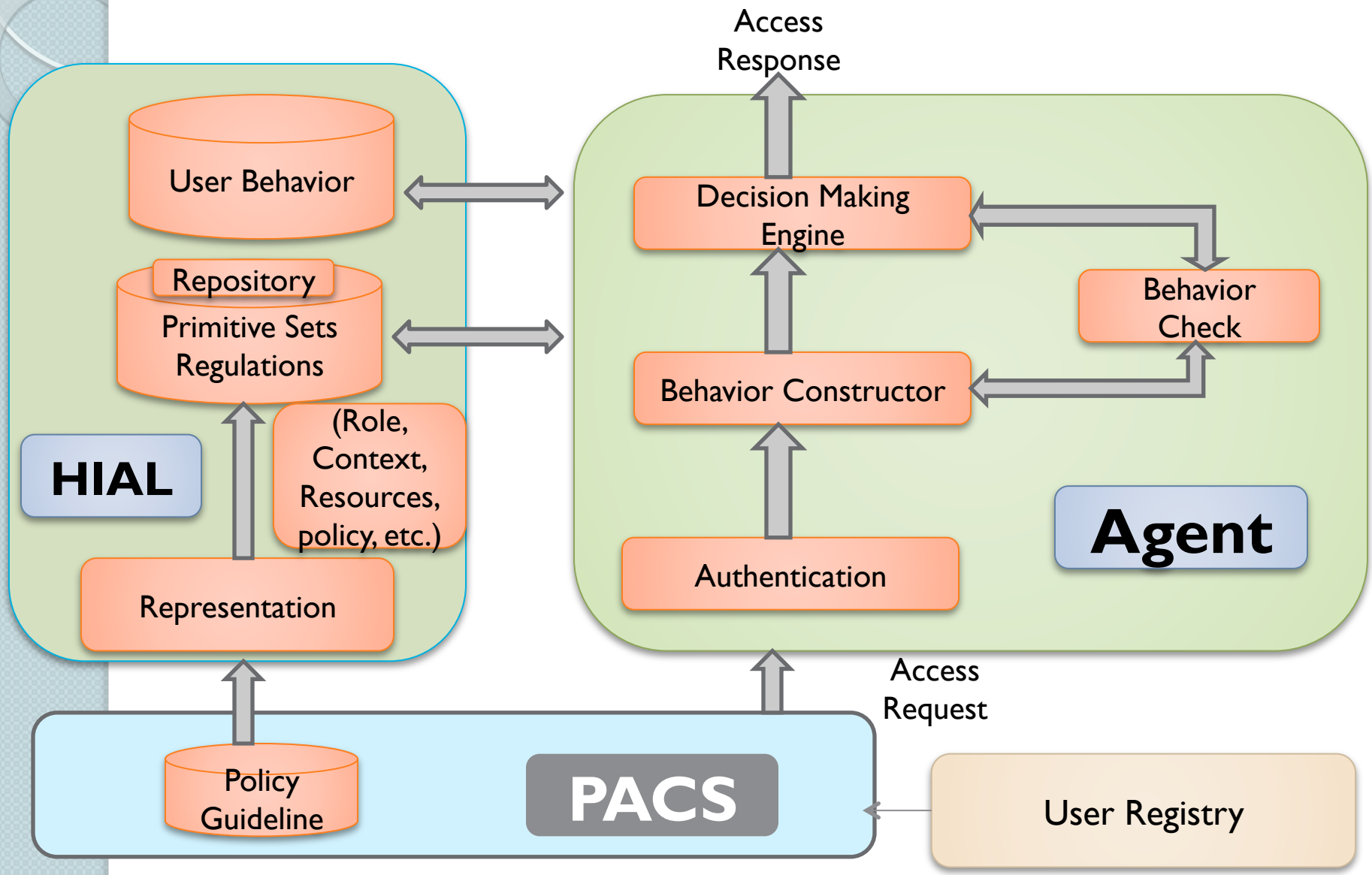
Stage 1: PACS Authenticating with the user registry to use the designed infrastructure



Stage 2 : Agent managed behaviour based access control infrastructure



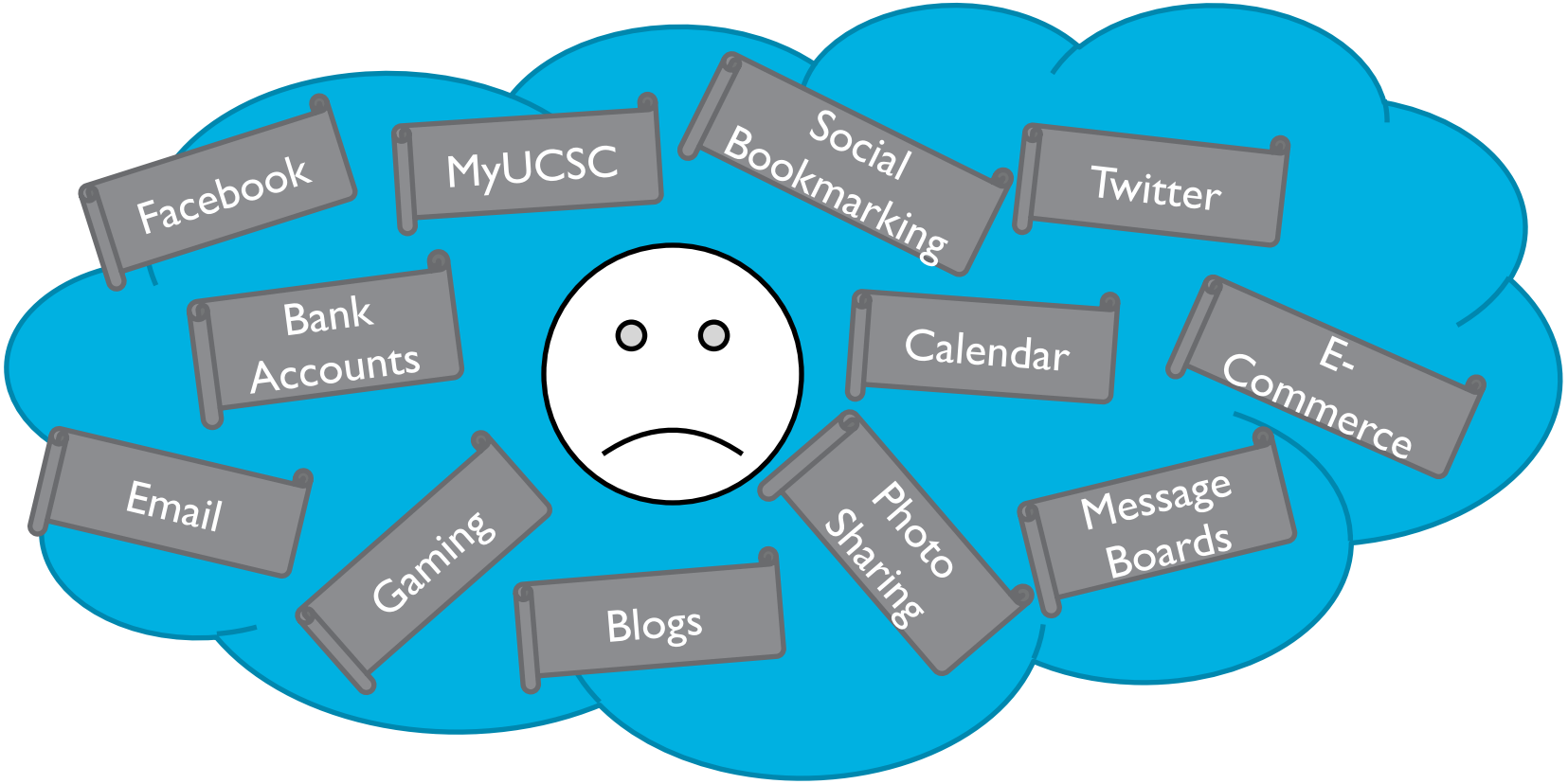
Complete Architecture



Introduction to OpenID

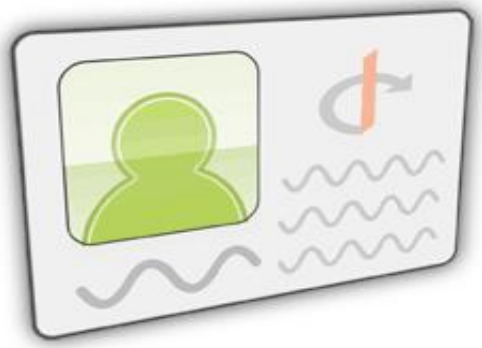
Need for OpenID

- Lots of websites, lots of accounts…

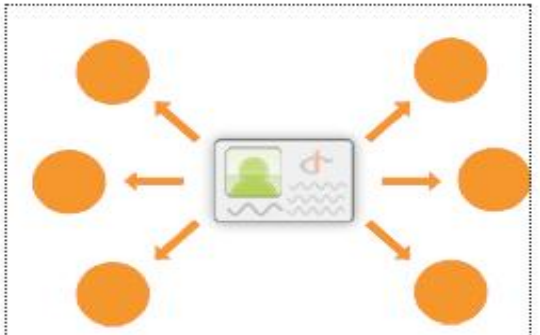


OpenID Solution

- Use one identity for all the internet service (OpenID enabled)



OpenID is a free and easy way to use a **single digital identity** across the Internet.



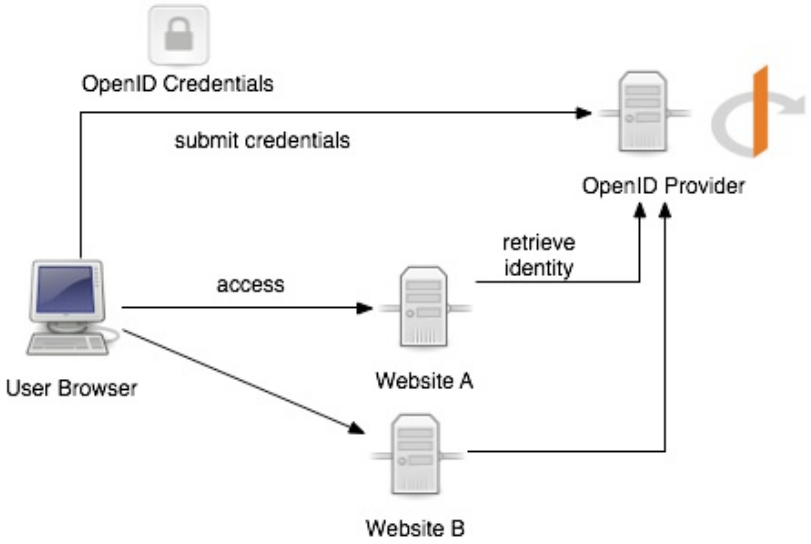
With one OpenID you can login to all your **favorite websites** and forget about online paperwork!

An OpenID is a URL

- URL are Globally unique.
- OpenId allows proving ownership of an URL
- People already have identity at URLS via blogs, photos, Myspace and Facebook Etc

Main Components

- **End-user**
  - The person who assert his or her identity to a site.
- **Identifier**
  - The URL chosen by the end-user as their OpenID identifi
- **Identity provider or Open provider**
  - A service provider offering t service of registering OpenII URLs
  - E.g. Yahoo, Blogger, etc
- **Relying party**
  - Site that wants to verify the end-user's identifier : "service provider".



Website Benefits

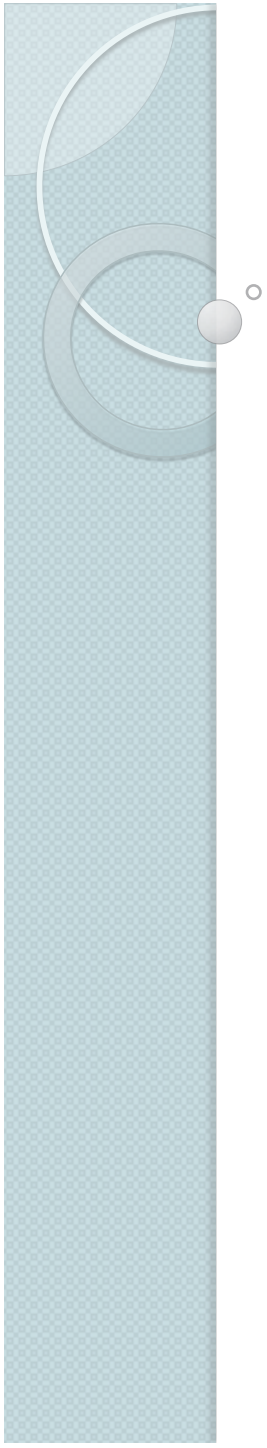
- Increased conversion rates from "site visitors" to "registered users"
- Reduced customer care cost and frustration with forgotten passwords
- Accelerated adoption of "community" features
- Limited password sharing issues
- Facilitated single sign-on across multiple company and partner websites

User Benefits

- Faster & easier registration and login
- Reduced frustration from forgotten user name/password
- Maintain personal data current at preferred sites
- Minimize password security risks

Challenges

- Though you have one, there are not many places to use it (yet) None of the big players — AOL, MS, Google, Yahoo!, MySpace — accept OpenID
- The sign-in process can be very confusing and jarring to users
- Security Concerns have not been fully resolved : subject to phishing attacks
- Unrealized loss of Anonymity



Introduction to OAuth

Function of OAuth

"OAuth provides a way to grant access to your data on some website to a third website, without needing to provide this third website with your authentication information for the original website."

oAuth Overview

- Security protocol that allows users to grant third-party access to their web resources without sharing their passwords.
- The heart of OAuth is an authorization token.
- OAuth is an open protocol
- Manages handshake between applications
- Used when an API publisher wants to know who is communicating with the system.

OAuth terminology

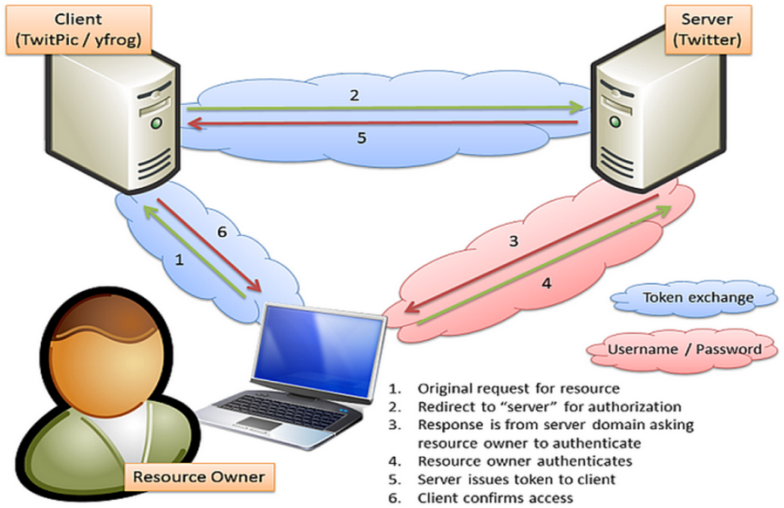
- **The resource owner** (original OAuth name: user) – that's you, me, or anyone with something private they want to share

- **The server** (original OAuth name: service provider) – that's the service where the private resources reside

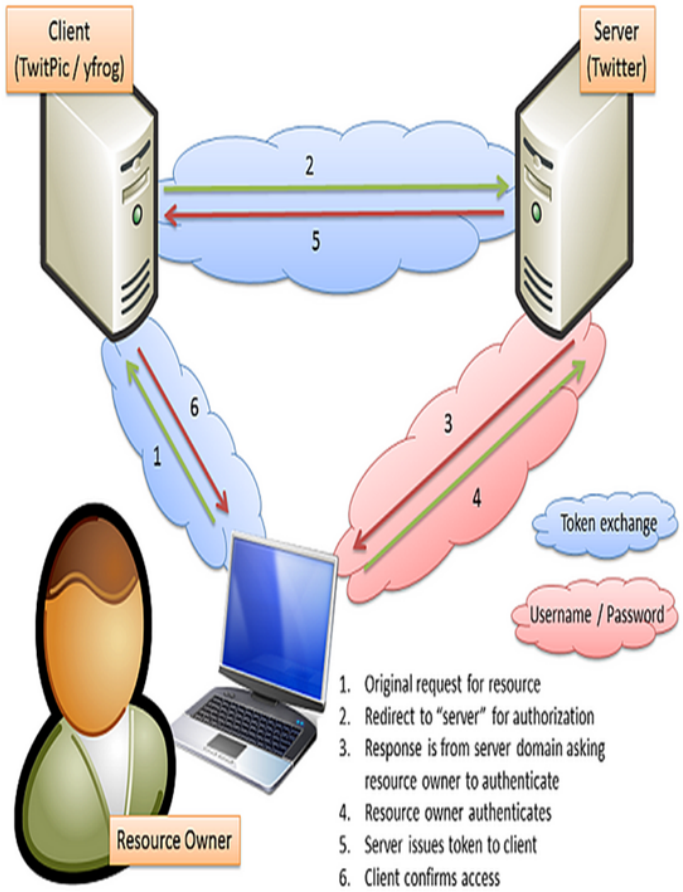
- **The client** (original OAuth name: consumer) – that's the service we'd like to use. It needs access to the resources

Example Scenario

- User has Twitter account and he wants to use a service such as TwitPic or yfrog to upload a photo and tweet it.
- Twitter account (or spe actions on twitter account reading, posting etc) is private resource and it shoulc protected



- Resource owner has t authorise the client (TwitPic o yfrog) to access protecte resources (twitter API actions on the server.
- Client asks the server t authenticate
- User grant or deny access t specific resources on the server
- Client is issued with a token tha can be presented to the serve to access those resources i future.



Case Study

E-health Services with Secure Mobile Agent

Rossilawati Sulaiman, Xu Huang, Dharmendra Sharma
Department of Information Science & Engineering
University of Canberra
Australia

Main Focus

" How Sender can securely transfer sensitive information to Recipient while still maintaining control over it "

- Introduces **mobile agents** to Multilayer Communication (**MLC**) layer in the model
- Sender keeps the **key** for decryption at his/her side until the agent needs it
- A **token** is carried by the agent to obtain the key for decryption processes

Main Components

Security Token

- It is an encrypted random number carried by the mobile agent to the Recipient's host
- Agent sends back the token to the Sender to retrieve the information for data decryption

Security mechanisms

Data Security

Protect the database from unauthorized access

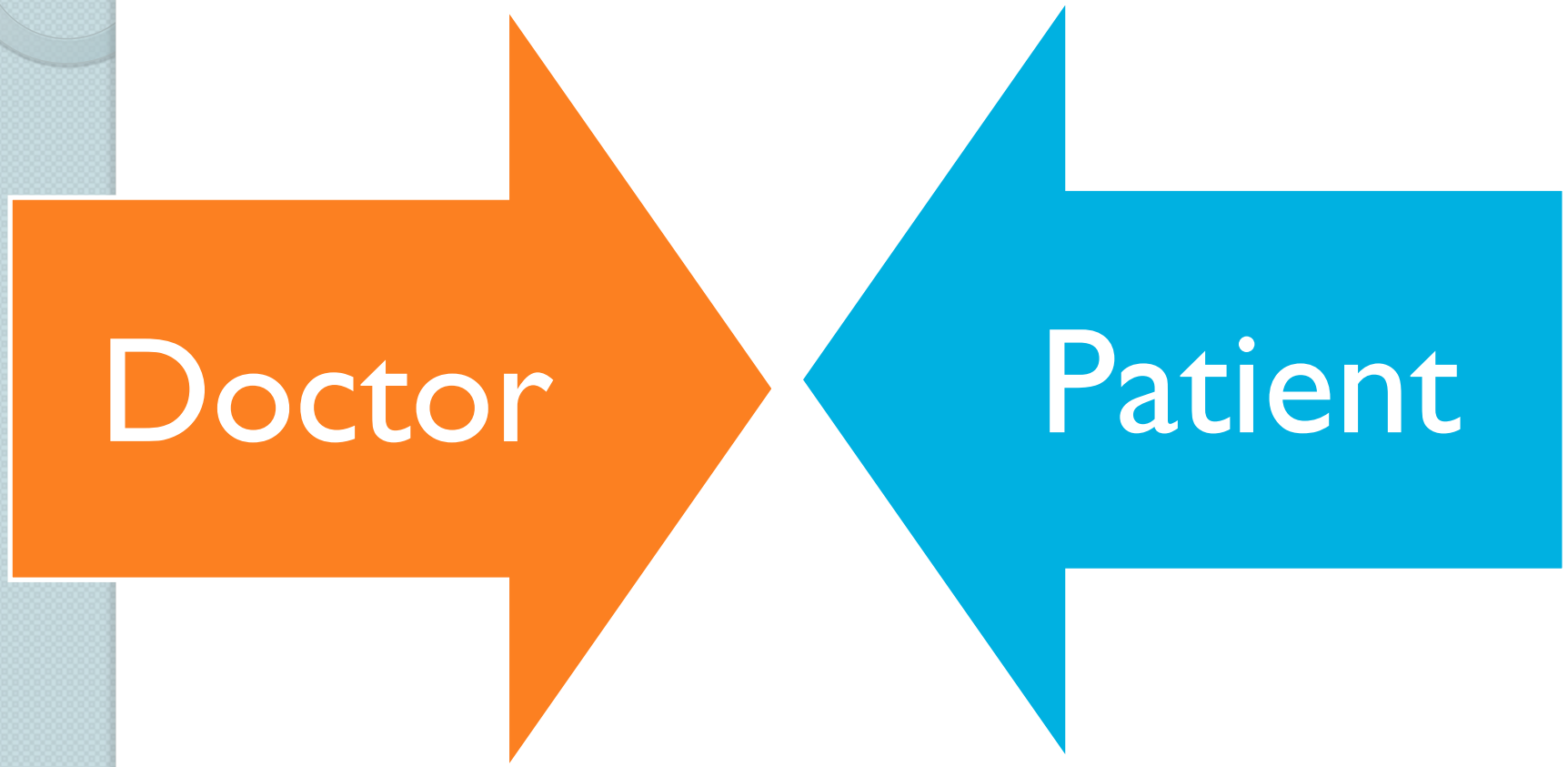
Channel security

Ensures security of a given communication channel, regardless of the information that is transferred over that channel

Classification and Security Mechanisms in the MLC

Layer of communication	Security Mechanism
Layer 1 : Extremely sensitive data Doctor → Doctor Doctor → Patient Doctor → Nurse Nurse → Patient	Data and Channel security
Layer 2 : Highly sensitive data Paramedic → Sys Coordinator	Data security (using wireless network)
Layer 3 : Medium sensitive data	Channel security or Data security
Layer 4 : Low sensitive data	Channel security or Data security
Layer 5 : Non sensitive data or public data The public	Secure open channel , ID and password

Example Scenario : Communication between Doctor and Patient



Steps involved

Step 1

Layer of communication (com_layer) is identified

Step 2

Choosing the appropriate security mechanism

Lo Value to choose the MLC layer

Role	Lo Value
Doctor Patient Nurse	Layer 1
Paramedic Coordinator System Coordinator	Layer 2
Social Worker	Layer 3
System Administrator	Layer 4

Finding com_layer value

Lo Value :	Com_layer Value
Sender = Recipient	Sender's L0 / Recipient's L0
Sender > Recipient	Sender's L0
Sender < Recipient	Recipient's L0

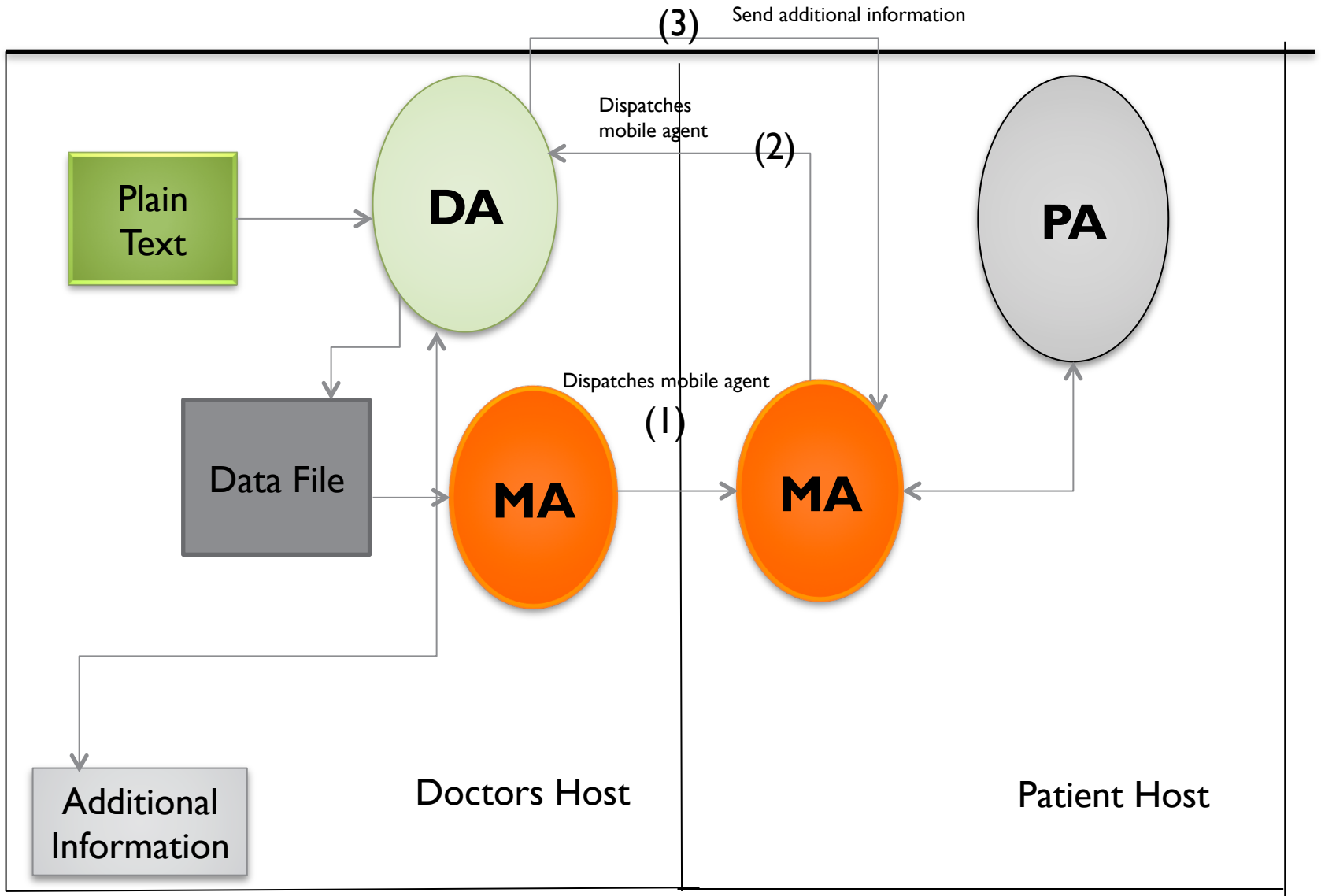
Appropriate Layer and Corresponding Security mechanism

Role	Lo Value
Doctor Patient Nurse	Layer 1
Paramedic Coordinator System Coordinator	Layer 2
Social Worker	Layer 3
System Administrator	Layer 4

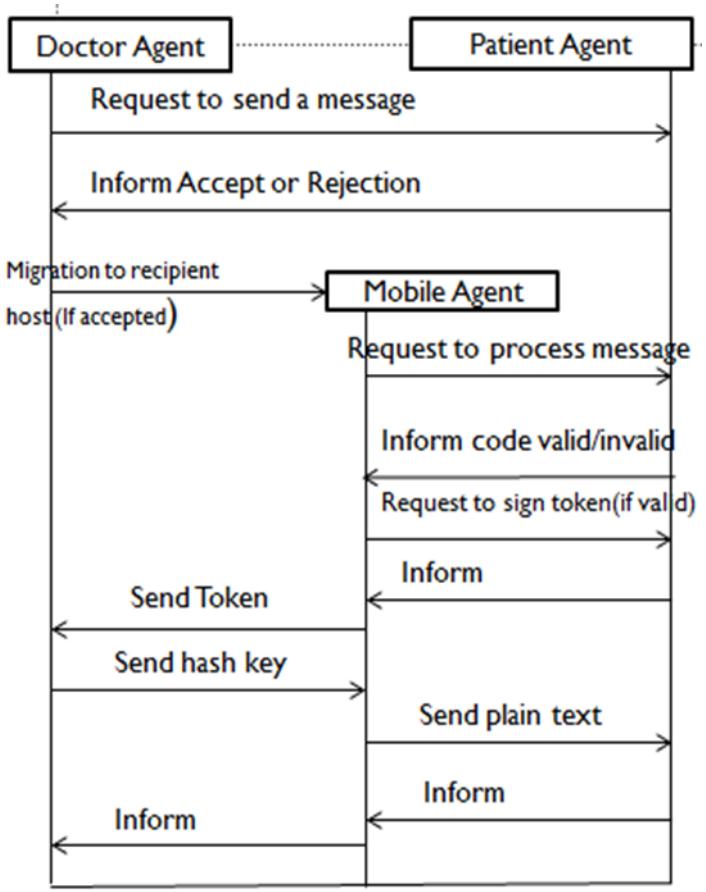
Layer of c
Layer 1 : Extr Doctor → Doctor Doctor → Pati Doctor → Nur Nurse → Patie
Layer 2 : High Paramedic → S
Layer 3 : Med
Layer 4 : Low
Layer 5 : Nor public data The public

Lo Value :	Com_layer Value
Sender = Recipient	Sender's L0 / Recipient's L0
Sender > Recipient	Sender's L0
Sender < Recipient	Recipient's L0

Security Architecture



Process flow



Conclusion

- Research implements a common infrastructure for secure sharing between PACS and the diagnostic image repository of EHR

- Agent based methodology can be used to implement this solution in the HIAL layer of EHR

Thank You

&

Questions?