

SUPPORTO A REGOLE CHIMICO-SEMANTICHE PER LA COORDINAZIONE DI PERVASIVE SERVICE ECOSYSTEMS

Tesi di Laurea Specialistica

Presentata da:

MATTEO DESANTI

Relatore:

Prof. MIRKO VIROLI

Correlatori:

Ing. ELENA NARDINI

Ing. DANILO PIANINI

Outline

- ❧ Ecosistemi Pervasivi
- ❧ Progetto SAPERE
- ❧ Scenari applicativi
- ❧ Obiettivi della tesi
- ❧ Prototipo di Infrastruttura: Architettura
- ❧ Modello e linguaggio delle Eco-law
 - ❧ Rappresentazione astratta
 - ❧ Esempi di Eco-law dalle Demo
- ❧ Conclusioni e sviluppi futuri

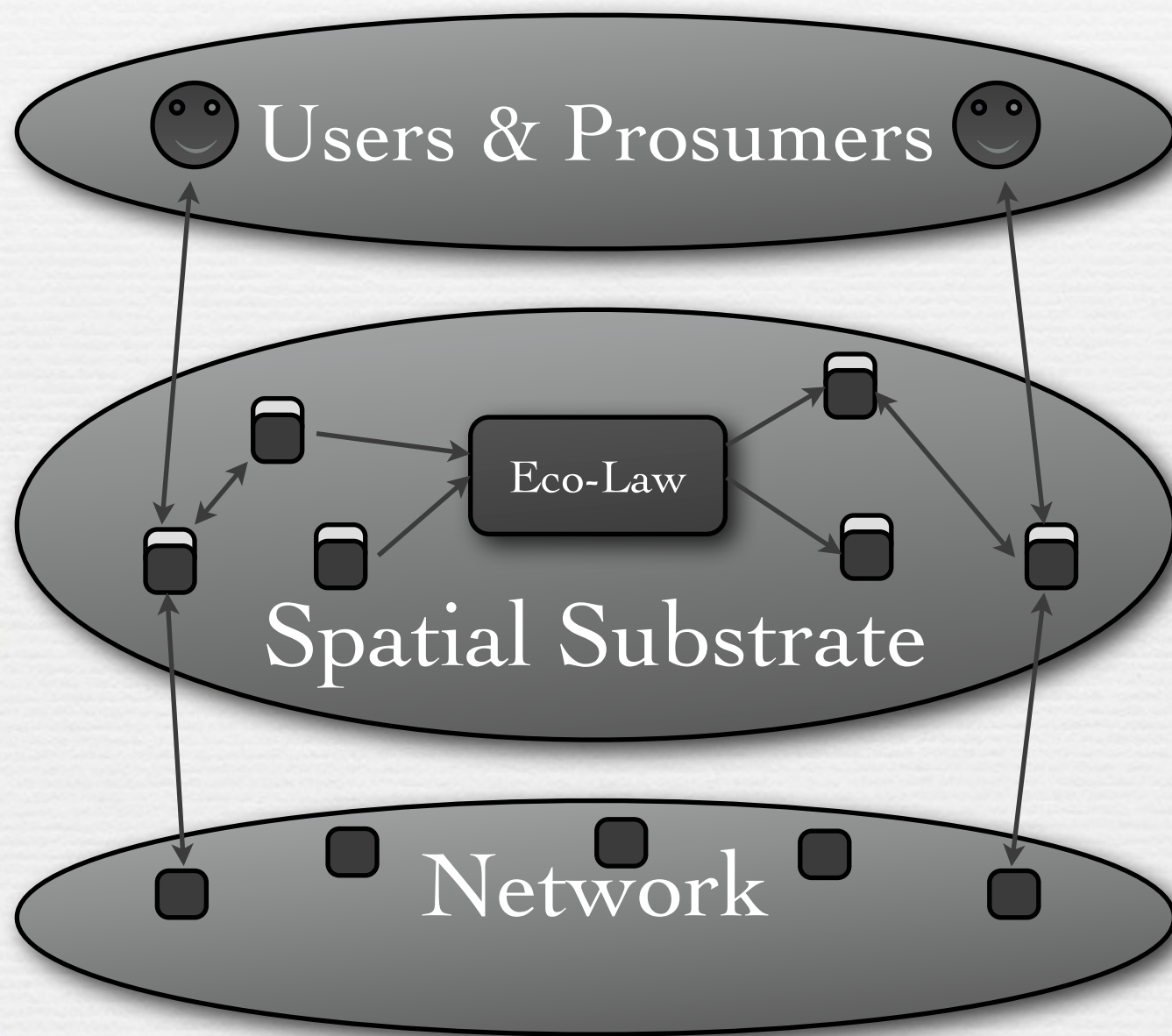
Ecosistemi pervasivi

- ❧ Paradigma promosso dalla diffusione di device portatili dotati di potenza computazionale sempre maggiore.
- ❧ Integrazione della gestione ed elaborazione delle informazioni in attività ed oggetti di uso comune.
- ❧ Funzionalità complesse realizzate collettivamente, coordinando i vari componenti.
- ❧ Sistemi in evoluzione continua, servizi e informazioni di tipo eterogeneo.
- ❧ Requisiti di fault tolerance, auto-organizzazione, adattabilità.

Progetto SAPERE

- ❧ Progetto STREP finanziato dalla UE all'interno del FP7.
- ❧ Obiettivi:
 - ❧ Sviluppare un framework per la gestione di servizi in scenari di pervasive computing.
 - ❧ Adottare un modello innovativo di coordinazione.
 - ❧ Radicalmente diverso dalle attuali architetture orientate ai servizi.
- ❧ Sfrutta astrazioni di ispirazione chimico-biologica.

Struttura Astratta



- Substrato decentralizzato inserito tra utenti ed infrastruttura di rete.
- Spazio Linda-like distribuito.
- Gestisce informazioni prodotte e consumate da utenti e servizi.
- Meccanismi di coordinazione nature-inspired realizzati da Eco-law.
- Componenti e servizi descritti da Live Semantic Annotations (LSA)

LSA

- ❧ Utilizzati per modellare indistintamente entità di tipo eterogeneo: utenti, servizi, informazioni, ...
- ❧ Formati da liste di proprietà multivalore.
- ❧ **Live:** costantemente modificate durante la loro esistenza all'interno del sistema.
 - ❧ Ecosistema dinamico e mutevole, in continua evoluzione.
- ❧ **Semantic:** descrivono le entità in termini concettuali, facendo riferimento ad un'ontologia.
 - ❧ Permette di ragionare sulle relazioni esistenti tra le entità.
- ❧ **Annotation:** semplici descrizioni logiche delle entità.

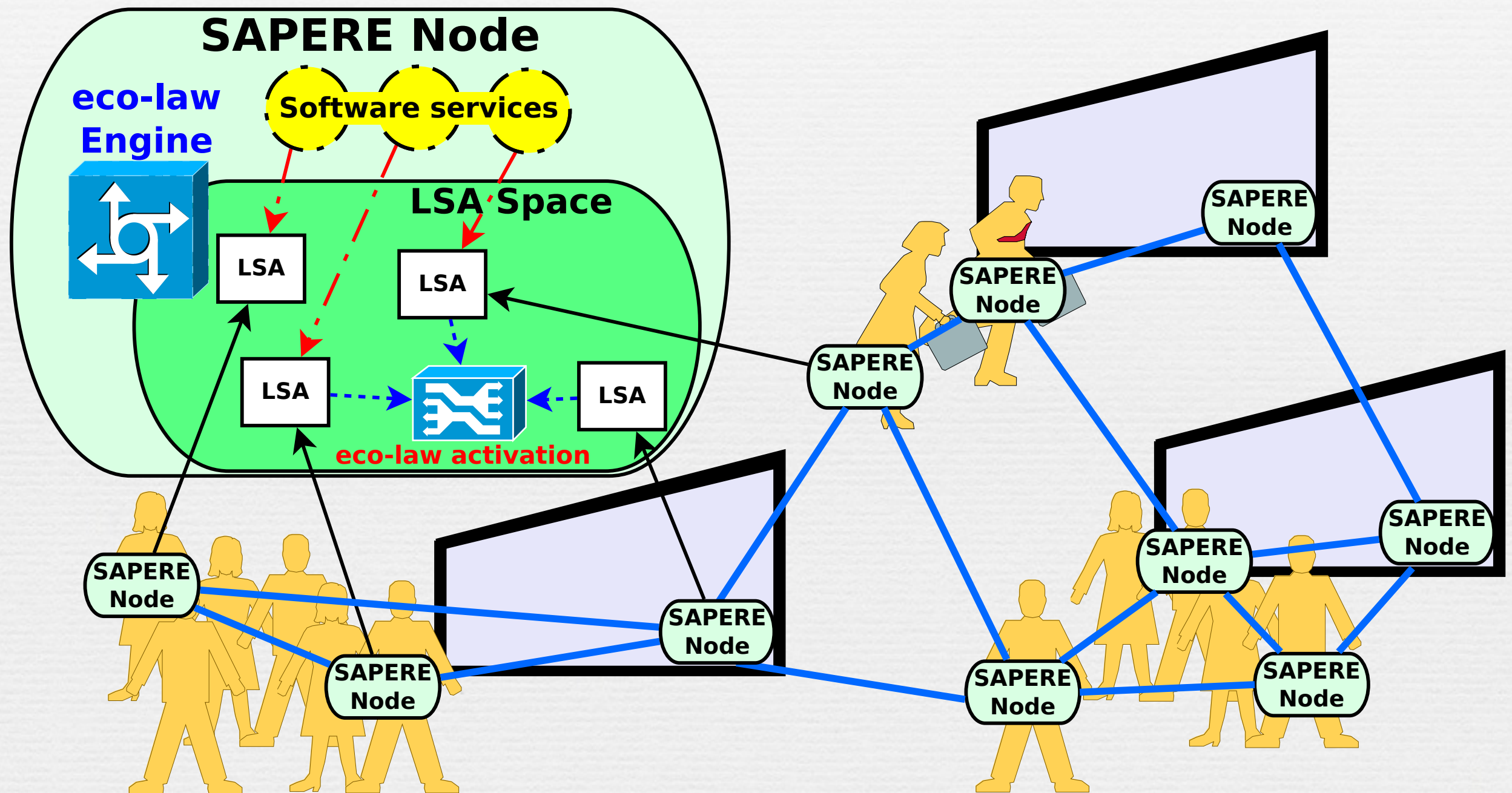
Eco-Law

`?USER:[type=(user); prefs has ?TOPIC] + ?CONT:[topic=?TOPIC]
-->`

`?RECOMMENDATION:[user=?USER; cont=?CONT]`

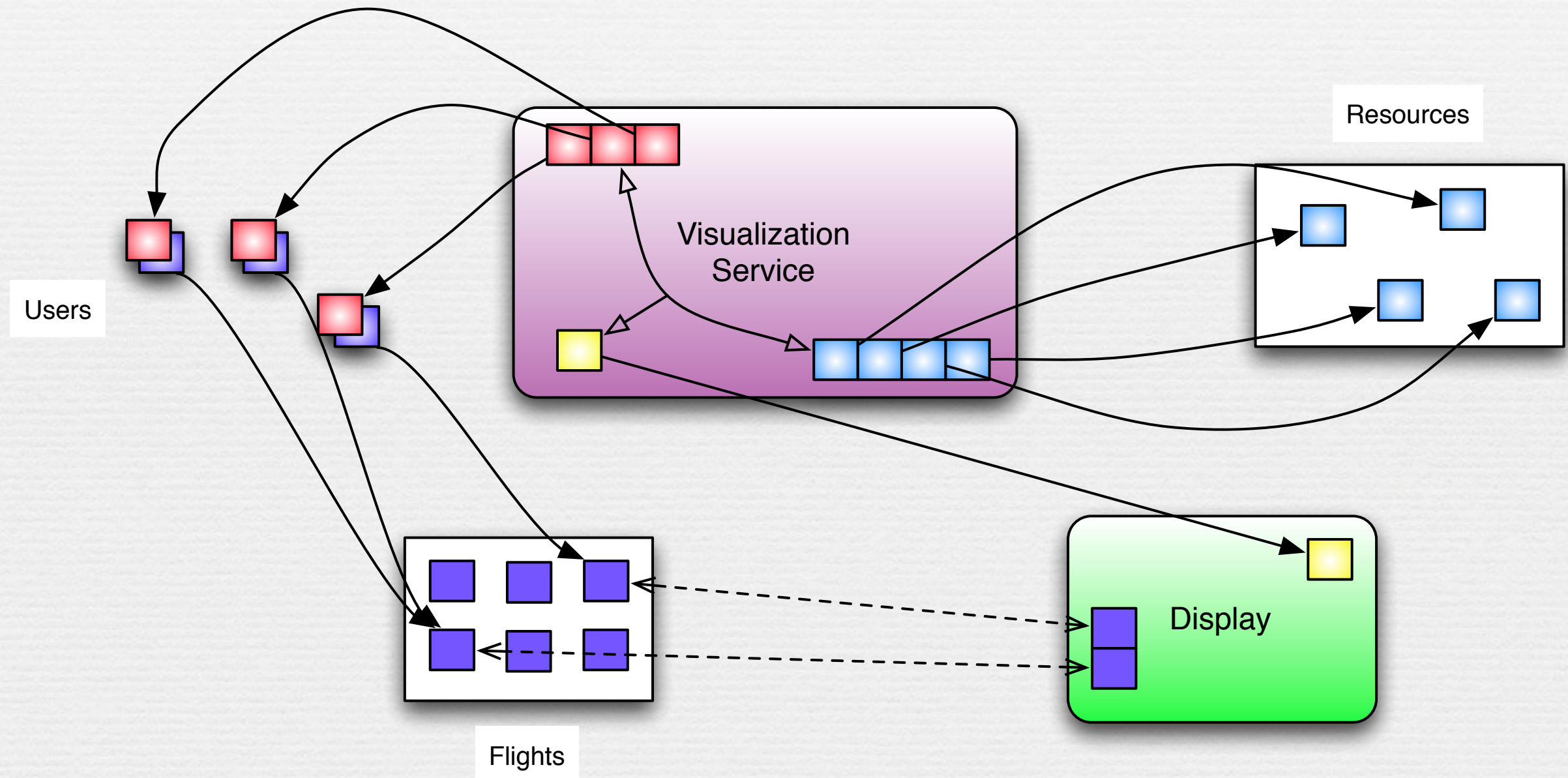
- Realizzano la coordinazione in modo auto-organizzante.
- Regole simili a reazioni chimiche, descritte tramite reagenti e prodotti.
 - Gli LSA rappresentano l'equivalente delle molecole da combinare.
- Caratterizzate da un rateo di attivazione.
- I reagenti sono template parzialmente specificati riferiti agli LSA.
 - In fase di attivazione della Eco-law, vengono selezionati gli LSA candidati dallo spazio.
- I prodotti indicano gli effetti della Eco-law sullo spazio degli LSA.
 - L'attivazione di un Eco-law può comportare inserimento, modifica, rimozione di uno o più LSA

Topologia



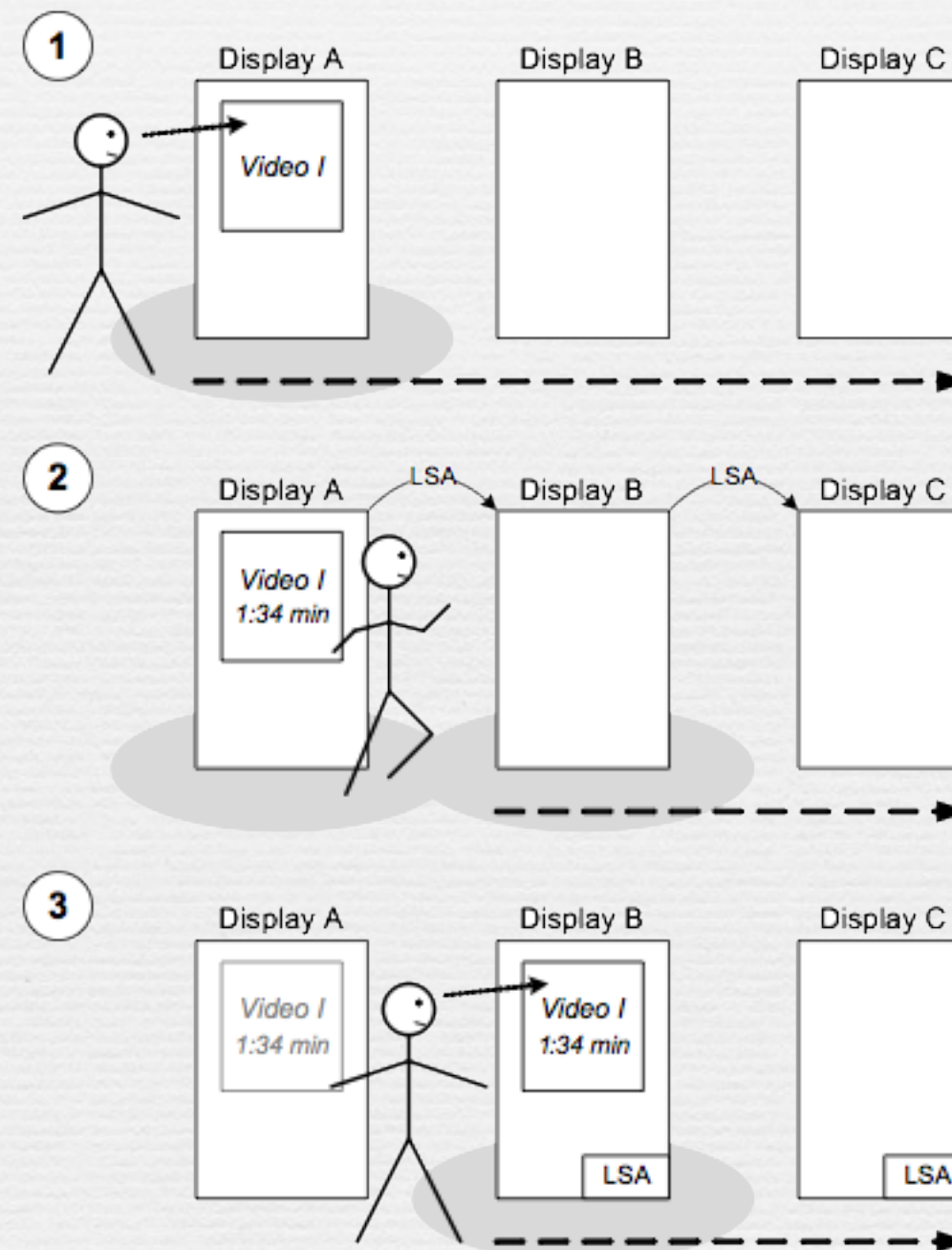
Scenari applicativi - 1

- Personalizzazione del contenuto in base al contesto



Scenari applicativi - 2

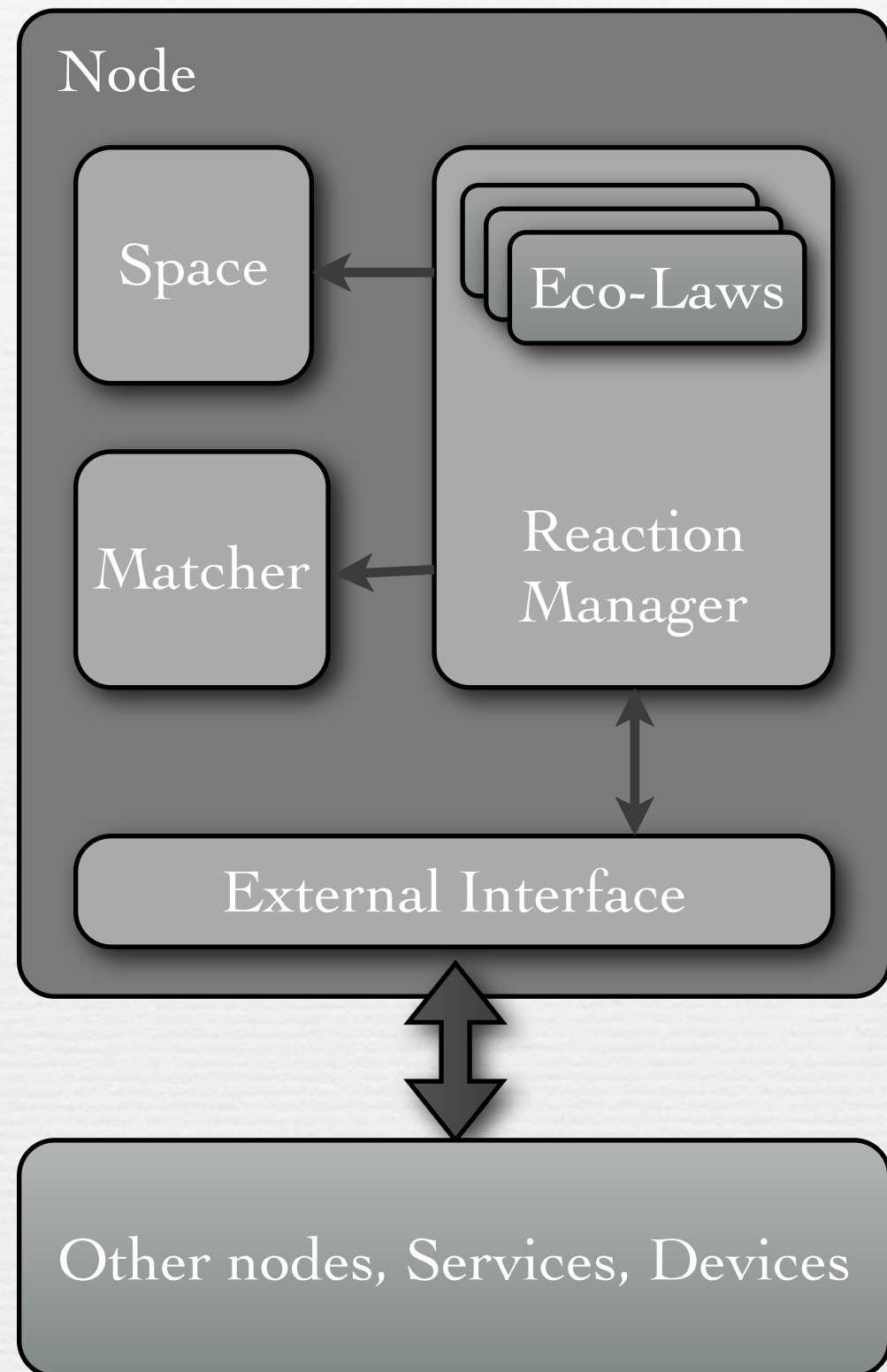
- Content that "follows" the user



Obiettivi della tesi

- Revisione completa del prototipo del progetto così da integrare completamente il nuovo modello di coordinazione basato su Eco-law
 - Mapping tra i costrutti definiti nella sintassi e le classi utilizzate nel prototipo Java.
 - Riprogettazione completa dei componenti adibiti alla esecuzione delle reazioni.
- Realizzazione di casi di studio non banali che ne sfruttano le funzionalità.

Prototipo: Architettura



- La topologia dell'infrastruttura rispecchia quella di rete.
- **Eco-Law**: modellano le regole di coordinazione dell'infrastruttura.
- **Reaction Manager**: gestisce le attivazioni delle eco-law.
- **Space**: repository per gli LSA locali.
- **Matcher**: valuta le associazioni tra reagenti ed LSA secondo quanto descritto nelle Eco-Law.
- **External Interface**: gestisce le comunicazioni del nodo con l'esterno.

Reaction Manager

- ❧ Gestisce una coda di eventi.
- ❧ Gli eventi rappresentano operazioni da effettuare sullo spazio.
 - ❧ Caratterizzati da una specifica collocazione temporale.
 - ❧ Un nuovo evento viene creato da attivazioni di Eco-law o da operazioni esterne, viste come reazioni immediate
- ❧ Ogni cambiamento del contenuto dello spazio degli LSA:
 - ❧ Comporta l'analisi delle Eco-law per selezionare quelle attivabili.
 - ❧ Può annullare l'attivazione di una Eco-law già schedulata.
- ❧ Usa un componente separato per il match e la valutazione.

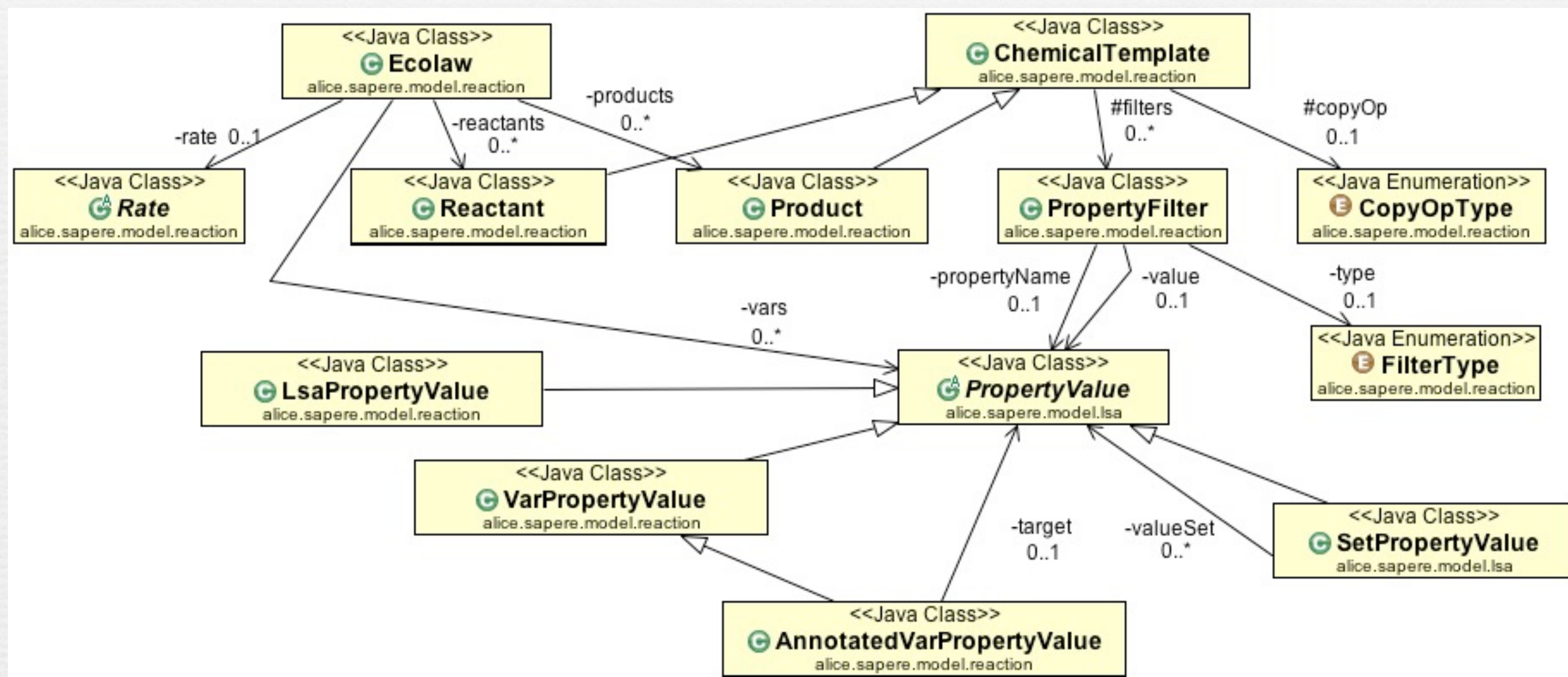
Reagent Matcher

- ❧ Selezionata una Eco-law ed una lista di LSA:
 - A. Viene scelto un LSA candidato per il primo reagente non analizzato.
 - B. Si effettuano le sostituzioni delle variabili e le associazioni necessarie.
 - C. Si riparte da A fino all'ottenimento di una istanza di Eco-law i cui parametri sono tutti completamente specificati.
- ❧ Per ogni Eco-law si valutano tutte le possibili combinazioni delle associazioni tra reagenti ed LSA.
- ❧ Alla Eco-law viene assegnato un punteggio basato sulla qualità del match tra reagenti ed LSA.
- ❧ Il punteggio totale influenza il rate della Eco-law.

Sintassi delle Eco-law

- ❧ Filtri: *termine Op termine*
- ❧ Ogni termine può essere rappresentato da:
 - ❧ Valori Ground singoli o in set (letterali)
 - ❧ Variabili
 - ❧ Variabili annotate (aggiungono vincoli)
- ❧ 3 tipi di operatori, semantica diversa per Reagenti e Prodotti
 - ❧ **=** : uguaglianza/assegnamento;
 - ❧ **Has** : appartenenza/aggiunta dell'elemento;
 - ❧ **Has-not** : assenza/rimozione dell'elemento.

Rappresentazione Astratta



Esempi dalla Demo 1:

[BOND-US]Collega l'LSA di un utente a un servizio di visualizzazione

```
?USER:[type=(user); preferences=?{PREF: ?PREF != ()}] +  
?SERV:[type=(visualization_service); users has-not ?USER]  
--->  
?USER + ?SERV:[users has ?USER]
```

- ❧ Il reagente USER seleziona utenti con preferenze non vuote
- ❧ SERV accetta dei servizi che non possiedono l'ID di USER
- ❧ Quando attivata, l'Eco-law collega i due LSA inserendo l'ID di USER nella proprietà “users” del servizio

Esempi dalla Demo 1:

[BOND-RS]Collega l'LSA di una risorsa a un servizio di visualizzaz

```
?RESOURCE:[type=(resource); topic=?{TOP: ?TOP != ()}] +  
?SERV:[type=(visualization_service); resources has-not ?RESOURCE]  
---->  
?RESOURCE + ?SERV:[resources has ?RESOURCE]
```

[VISUAL]Imposta il contenuto da visualizzare sul display

```
?SERV:[type=(visualization_service); res_to_vis=?RESOURCE] +  
?DISP:[type=(display); vis=(ready); resource has-not ?RESOURCE] +  
?RESOURCE  
---->  
?DISP:[vis=(on); resource has ?RESOURCE] + ?SERV + ?RESOURCE
```

Esempi dalla Demo 2:

[PRE-MOVE] Ferma il display quando l'utente si allontana

```
?USER:[type=(user); position=?POSITION] +  
?POSITION:[type=(localization); signal=?{SIG: ?SIG < -60}] +  
?DISPLAY:[type=(display); vis=(on); user=?USER]  
--->  
?USER + ?POSITION + ?DISPLAY:[vis=(idle)]
```

[HIST] Crea e diffonde lo storico dell'utente

```
?DISPLAY:[type=(display); vis=(paused); user=?USER; content=?CONT;  
time=?TIME] +  
?NEIGH:[type=(#neighbour); where=?LOC]  
--->  
?DISPLAY:[vis=(ready); user=(); content=(); time=(0.0)] + ?NEIGH +  
?HISTORY1:[type=(history); time=?TIME; user=?USER; content=?CONT] +  
?HISTORY2:[type=(history); #location=?LOC; time=?TIME; user=?USER;  
content=?CONT]
```

Conclusioni e sviluppi futuri

- ❧ Il sistema supporta completamente il modello delle Eco-law
- ❧ Estendere e migliorare l'infrastruttura.
 - ❧ Gestione ottimizzata delle reazioni.
 - ❧ Grafi delle dipendenze tra LSA e Eco-law.
 - ❧ Più fasi di matching in parallelo.
 - ❧ Matching semantico tra reagenti ed LSA.
- ❧ Sviluppo di un interprete per il linguaggio delle Eco-law.
- ❧ Modelli e generatori di codice per facilitare lo sviluppo degli agenti e l'interazione con l'infrastruttura.

La parola al mio collega
Michele Morgagni.

MODULO DI COMUNICAZIONE IN UNA INFRASTRUTTURA PER PERVASIVE SERVICE ECOSYSTEMS

Tesi di Laurea Specialistica

Presentata da:

MICHELE MORGAGNI

Relatore:

Prof. MIRKO VIROLI

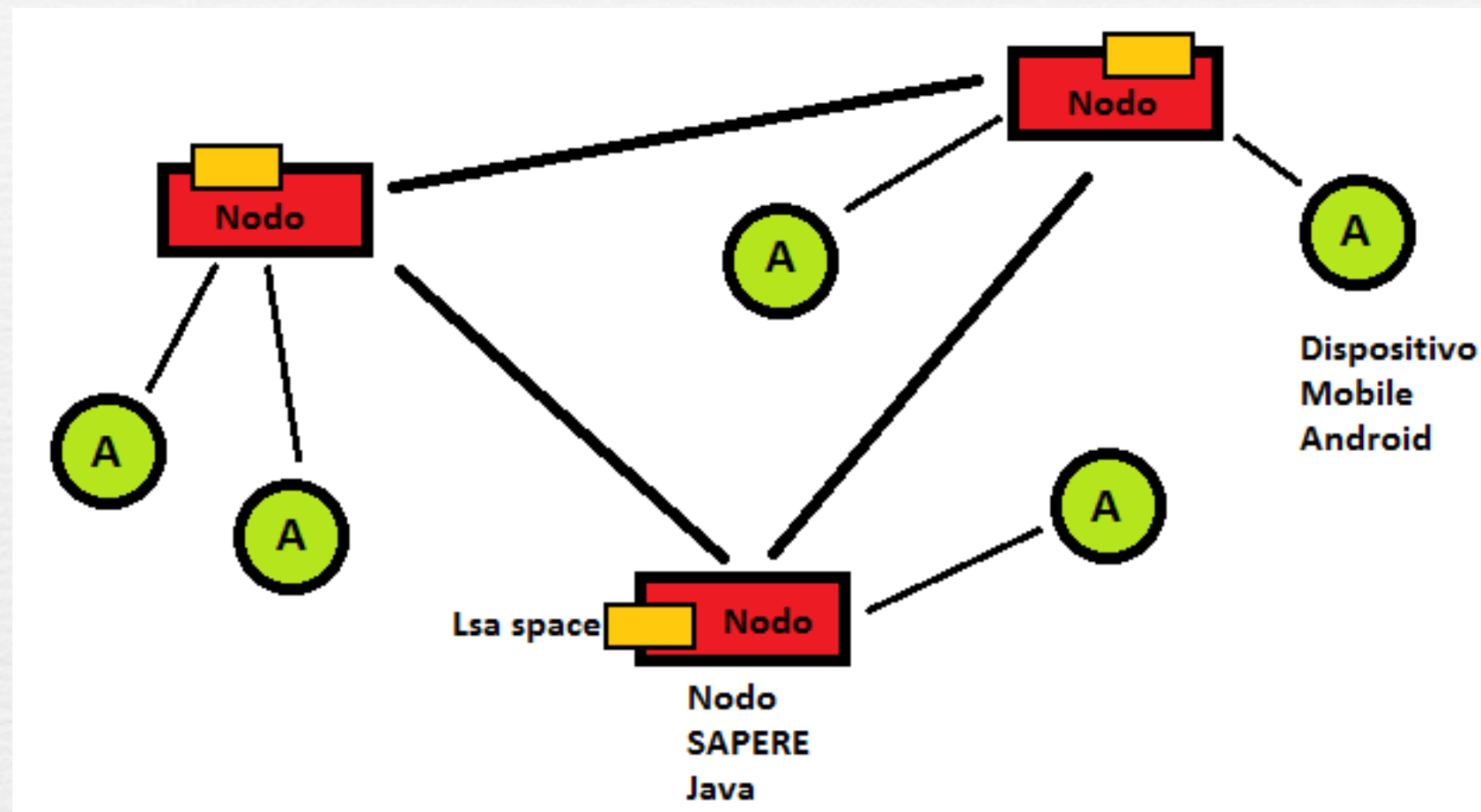
Correlatori:

Ing. DANILO PIANINI

Outline

- ❧ Scenario
- ❧ Requisiti
- ❧ Obiettivi
- ❧ Analisi dei protocolli wireless
- ❧ Progettazione dei dispositivi mobili
- ❧ Progettazione dei nodi
- ❧ Casi di studio
- ❧ Conclusioni e sviluppi futuri

Scenario



- ❧ Nodi SAPERE in ambiente Java
- ❧ Dispositivi mobili degli utenti in ambiente Android
- ❧ Comunicazioni wireless tra i dispositivi

Requisiti

- ❧ Sistema distribuito
- ❧ Comunicazioni wireless tra i dispositivi
- ❧ Alta dinamicità ed adattatività del sistema
- ❧ Supporto a scenari con un numero di utenti elevato
- ❧ Possibilità di realizzare i molteplici casi di studio
SAPERE
- ❧ Integrazione con parte del sistema pre-esistente

Obiettivi

- ❧ Realizzazione della piattaforma di comunicazione e di alcuni casi di studio
 - ❖ Lato Android:
 - ◆ Libreria ed interfaccia per l'accesso a SAPERE
 - ◆ Esempio di applicativo per l'uso della libreria
 - ❖ Lato Java:
 - ◆ Agenti per la comunicazione tra nodi
 - ◆ Agente per la gestione dei dispositivi mobili connessi
 - ❧ Esempi di agenti case-study dependent

Analisi dei protocolli wireless (1/3)

☛ NFC (Near Field Communication)

- ❖ Corto raggio: 2-5 cm
- ❖ Nessun supporto in ambiente Android (uscita prevista)
- ❖ Possibile utilizzo: autenticazione dell'utente al primo accesso nel sistema

☛ Bluetooth

- ❖ Raggio variabile a seconda delle antenne (1-100 metri)
- ❖ Basso consumo
- ❖ Creazione di Piconet con un massimo di 8 componenti
- ❖ Buon supporto sia in Java che in Android
- ❖ Possibile utilizzo: comunicazioni tra nodi o tra dispositivi mobili e nodi

Analisi dei protocolli wireless (2/3)

☛ Zig-Bee

- ❖ Raggio: 70-300 metri
- ❖ Basso consumo, bassa complessità
- ❖ Fino a 64000 nodi, con diverse tipologie di rete possibili
- ❖ Nessun supporto in ambiente Android (uscita prevista)

☛ Wi-fi

- ❖ Raggio: circa 100 metri, fino a 254 nodi in reti di classe C
- ❖ Possibili diverse modalità, supportati diversi protocolli di sicurezza
- ❖ Buon supporto in Android (no reti ad-hoc), gestione a livello di sistema dei nodi
- ❖ Possibile utilizzo: comunicazioni tra dispositivi mobili e nodi

Analisi dei protocolli wireless (3/3)

☛ Scelta effettuata:

☛ Wi-fi per comunicazione tra dispositivi mobile e nodi

- ❖ Buon supporto in Android
- ❖ Lato nodo configurato da sistema operativo
- ❖ Possibile stimare la distanza tra dispositivi attraverso la percezione della potenza del segnale
- ❖ Scala maggiore rispetto a Bluetooth

☛ Bluetooth per comunicazioni tra nodi

- ❖ Buon supporto in Java (librerie valide e funzionanti)
- ❖ Più reattivo rispetto a Wi-fi (una volta configurata la rete)
- ❖ Supporto all'utilizzo di più device BT contemporanei per nodo

Progettazione dei dispositivi mobili

Entità per l'accesso a SAPERE da dispositivo mobile

- ❧ Rappresentano l'LSA dell'utente
- ❧ PassiveLsa:
 - ❧ Entità passiva
 - ❧ Invia i messaggi al nodo più vicino
- ❧ ActiveLsa:
 - ❧ Entità attiva
 - ❧ Utilizza PassiveLsa
 - ❧ Nasconde il proprio comportamento
 - ❧ Si occupa autonomamente della sincronizzazione con il nodo più vicino
 - ❧ Supporta diverse politiche di sincronizzazione

```
<<interface>>
IPassiveLsa

+inject(content:Content): LsaId
+reinject(lsaId:LsaId,content:Content): boolean
+moveAway(lsaId:LsaId): boolean
+update(lsaId:LsaId,props:Property[]): boolean
+read(lsaId:LsaId): boolean
+observeOne(lsaId:LsaId): boolean
+observeMany(lsaId:LsaId): boolean
+deregister(lsaId:LsaId): boolean
+refresh(lsaId:LsaId): boolean
```

```
<<interface>>
IActiveLsa

+registerLsa(): boolean
+destroyLsa(): boolean
+setPolicy(policy:SyncPolicy): boolean
+getPolicy(): SyncPolicy
+getId(): LsaId
+setProperty(property:Property): boolean
+getProperty(name:String): Property
+getProperties(): Property[]
+addExternalLsa(id:LsaId): boolean
+deleteExternalLsa(id:LsaId): boolean
+getExternalProperty(id:LsaId,name:String): Property
+getExternalProperties(id:LsaId): Property[]
```

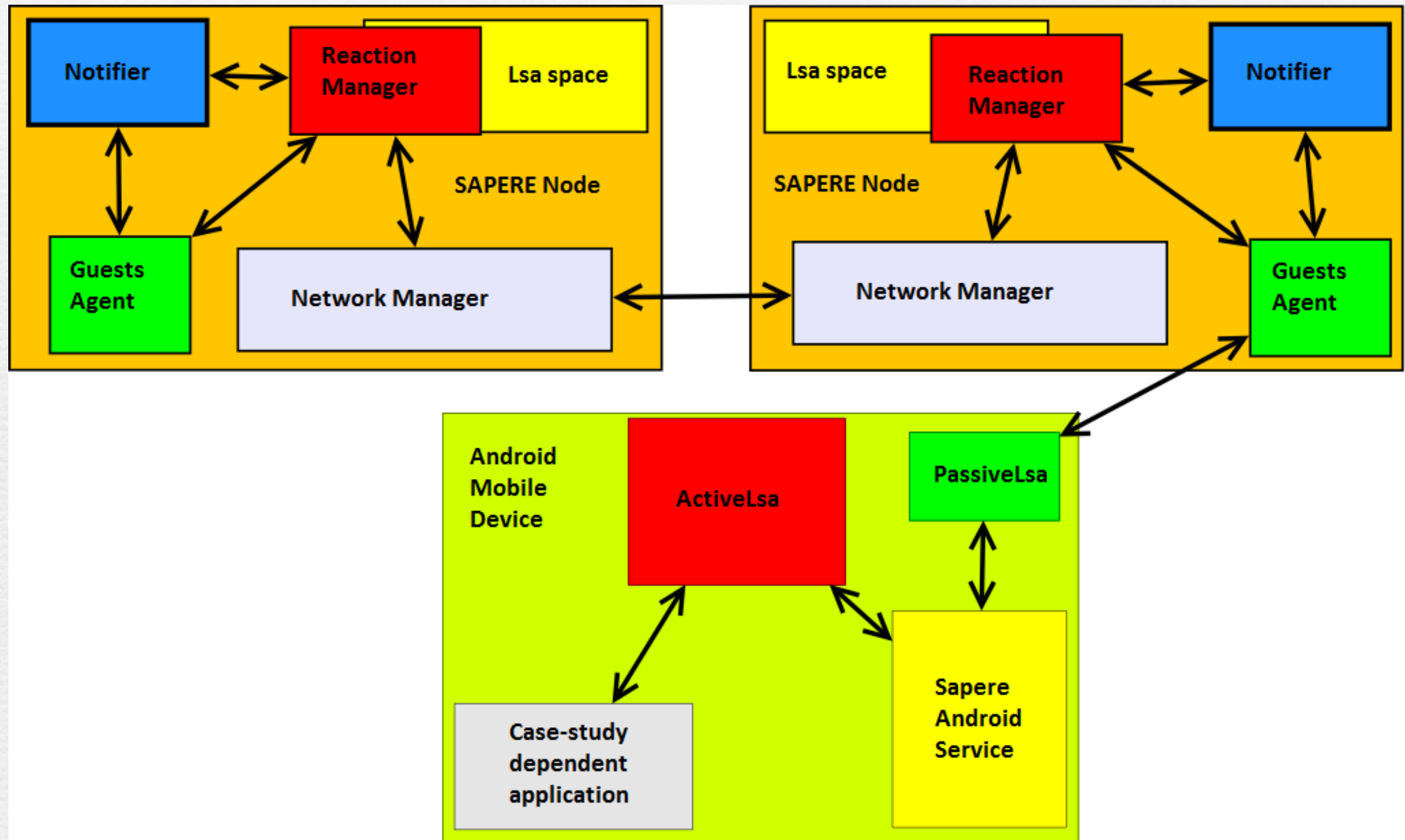
Comportamento ActiveLsa

- ❧ Ricerca periodica delle reti e selezione di quelle di interesse
- ❧ Connessione al nodo più vicino
- ❧ Migrazione tra nodi trasparente all'utente
- ❧ Invio periodico di informazioni per la localizzazione del dispositivo: potenza del segnale e orientamento
- ❧ Mantenimento di una copia consistente dell'LSA rispetto al sistema (attraverso richiesta di update e ricezione delle notifiche)

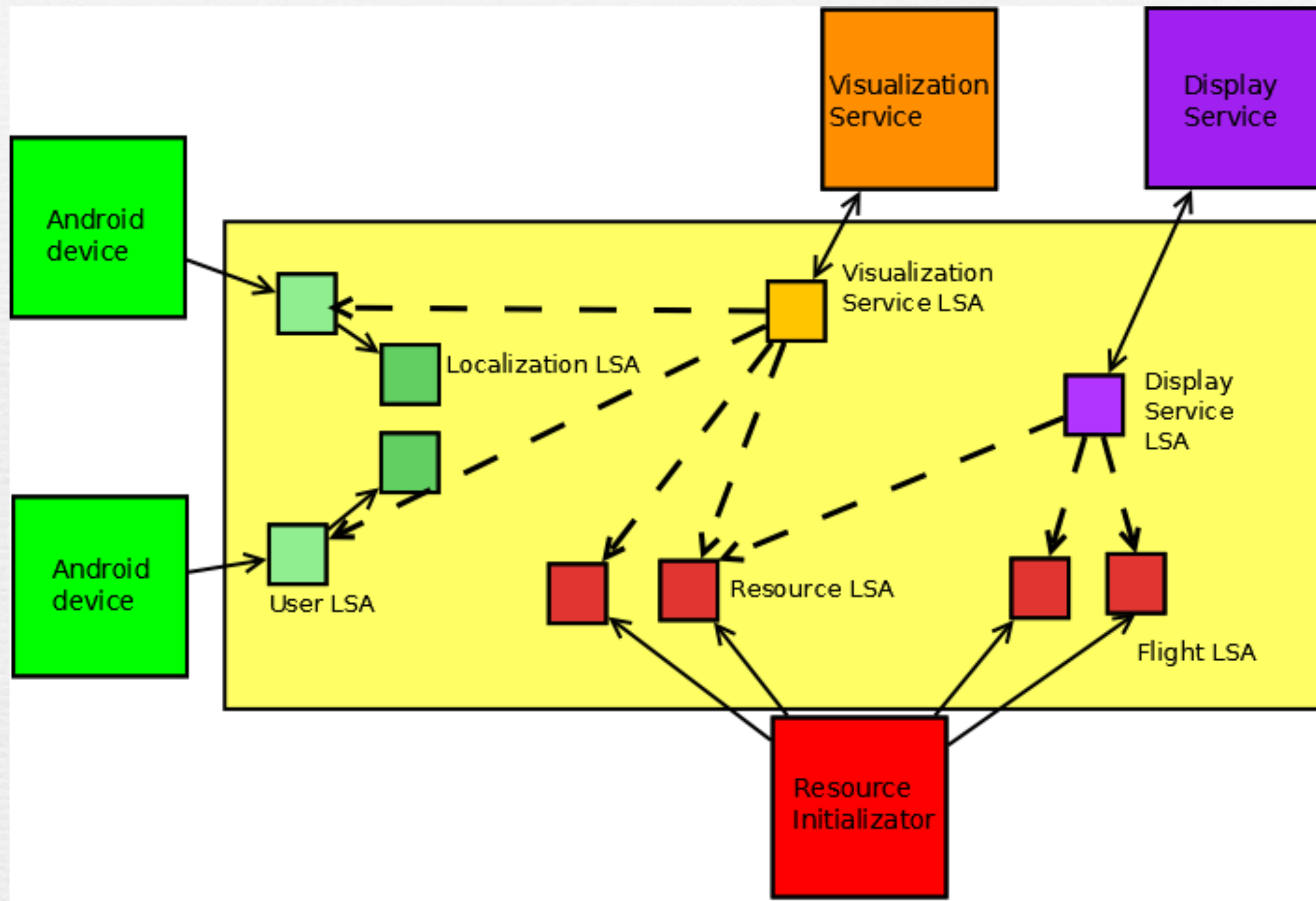
Progettazione dei nodi

- ❧ Gestione dei dispositivi mobili connessi
 - ❖ Creazione Access Point Wi-fi (sistema operativo)
 - ❖ Ricezione connessioni dai dispositivi mobili
 - ❖ Ricezione messaggi
 - ❖ Invio notifiche
 - ❖ Inserimento e modifica degli LSA corrispondenti
- ❧ Gestione dei nodi vicini
 - ❖ Ricerca dei nodi e scambio di informazioni iniziali tramite Bluetooth
 - ❖ Invio delle diffusioni
 - ❖ Predisposizione all'aggiunta di nuovi protocolli

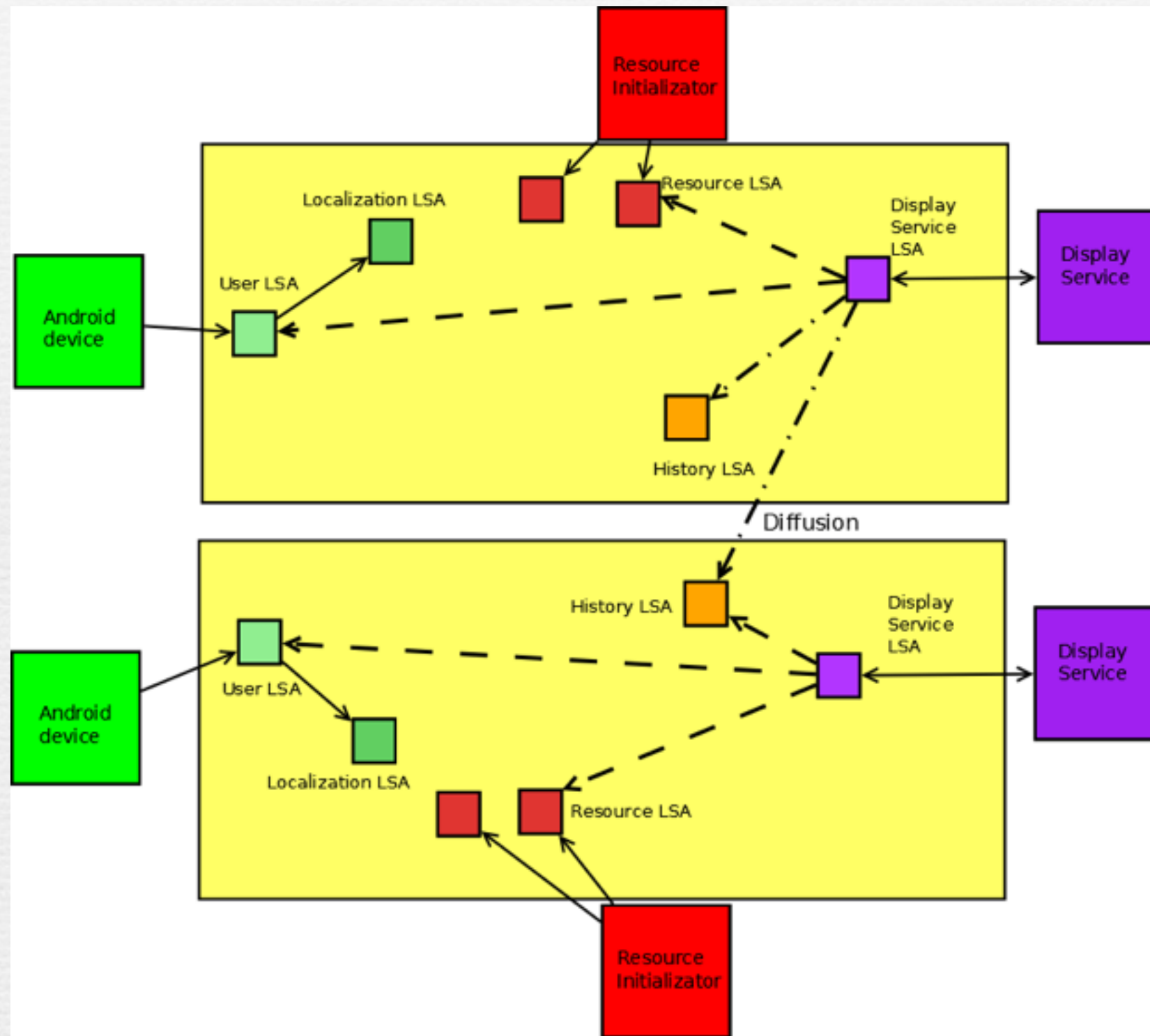
Scenario complessivo



Caso di studio 1



Caso di studio 2



Conclusioni e sviluppi futuri

❧ Attualmente:

- ❧ Prototipo distribuito di piattaforma funzionante
- ❧ Supporto alla realizzazione di casi di studio non banali

❧ Possibili evoluzioni:

- ❧ Aggiunta del supporto per le comunicazioni attraverso altri protocolli wireless
- ❧ Modifiche in funzione della disponibilità di nuove tecnologie abilitanti

DEMO

Grazie per
l'attenzione.