Ingegneria dei sistemi software e dei servizi in rete
A.A. 2016/2017

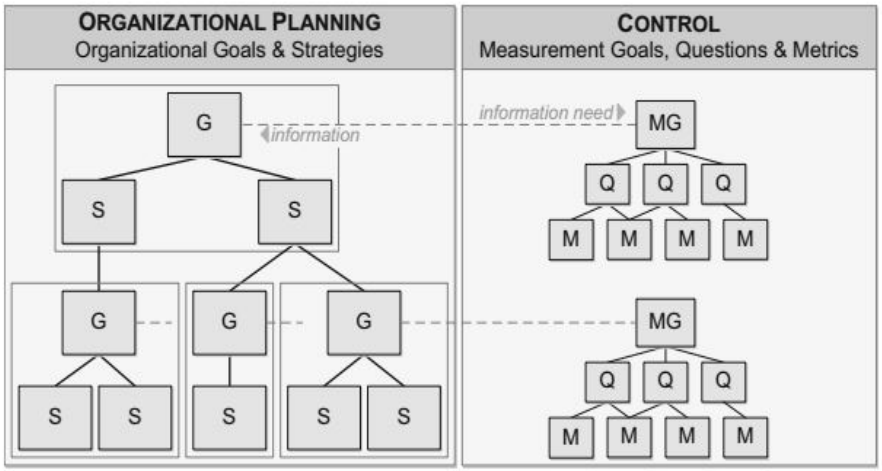
ERMES-QIP
Enterprise Service Bus
Estensione 2016

Marco Rosati
Alice Di Luise

Agenda

- Introduzione
  - Introduzione GQM + Strategies
  - Ambito e portata del progetto
  - Metodologia di sviluppo
- Enterprise Service Bus
  - Features
  - Casi d’uso
  - Enterprise Service Bus: Struttura messaggi
  - Spring integration
  - Architettura software
  - Implementazione
  - Conclusioni

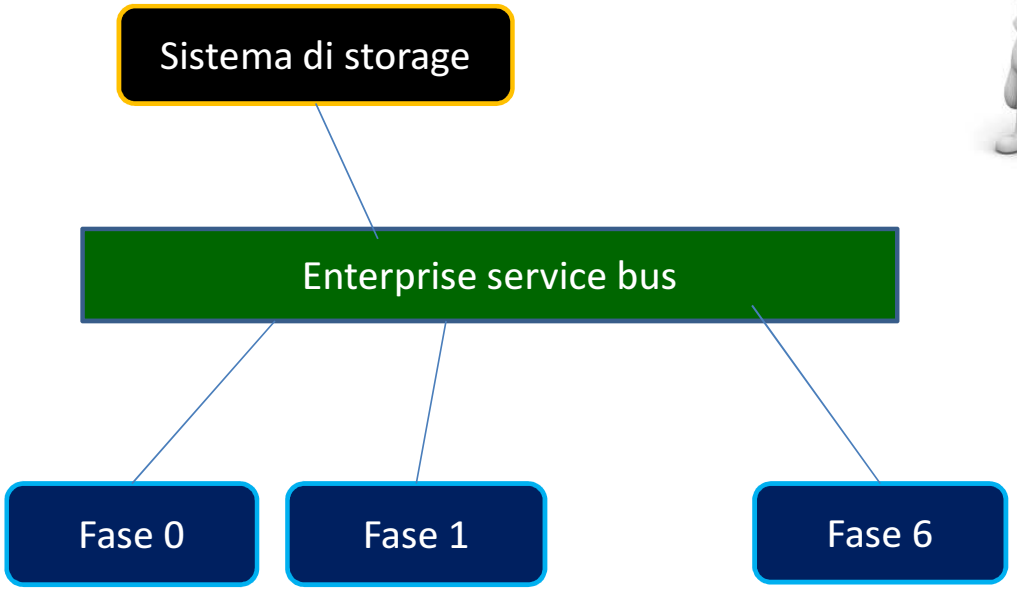
GQM+S: modello e processo



- Allineare goal e strategie tra unità organizzative
- Comunicare goal, strategie e misure all'interno dell'organizzazione
- Prendere decisioni basandosi su misure
- Monitorare in modo oggettivo il raggiungimento dei goal

ISSSR GQM+Strategies®

Ambito del progetto



-Fasi non allineate: codifiche e strutture diverse per gli elementi GQM+S
-Fase 2.1: processo di costruzione della griglia GQM+S parzialmente implementato.

.....

Metodologia di Sviluppo

- Progettazione e implementazione guidata dai casi d'uso
  - Documento di visione
  - Requisiti
    - User needs
    - Features
    - Casi d'uso
  - Glossario
  - Realizzazione dei casi d'uso

Documentazione esaustiva?
Interazione team fase 2.2 (configurazione, identificazione bug, test)

Enterprise Service Bus

Message transformation

Problema integrazione fasi 0-1-2.1, 2.2

- Gli elementi GQM+S creati dalle fasi 0, 1, 2.1 devono essere correttamente recuperati ed interpretati dalla fase 2.2
- Problemi:
  - Fase 2.1 esegue CRUD di elementi GQM+S come stringhe codificate in **base64**

    payload→ "eyJjcml…" typeObj→"base64-Assumption"
  - La fase 2.2 si aspetta di ottenere un oggetto **json**

    Payload→{id\":1,\"title\":\"A1 title\“,…} typeObj→”Assumption”
  - Le fasi non sono allineate relativamente alla struttura degli elementi GQM+S

    Assumption

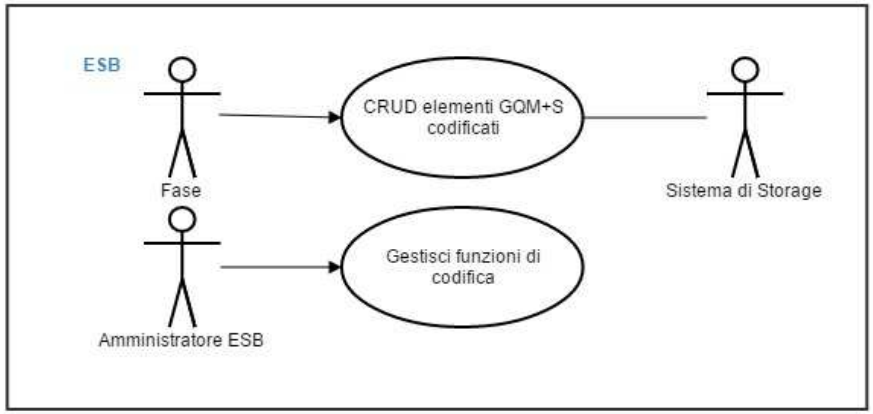
    status → validationState, projectId → instanceProjectId, formato creationDate ecc…

Features

ID	Titolo	Descrizione
F1	Leggi/cancella elementi GQM+S codificati	Il sistema permette ad un Fase di eseguire un'operazione di read/delete di uno o più elementi GQM+S presenti nello storage condiviso, specificando nel messaggio inviato tramite una richiesta http: • la codifica con cui si vuole ottenere l'elemento GQM+S. • campi necessari per l'operazione di read/delete (API ESB) • Le codifiche supportate sono: • Base64 • JSON • Custom
F2	Crea/aggiorna elementi GQM+S codificati.	Il sistema permette ad un Fase di creare/aggiornare un elemento GQM+S nello storage condiviso specificando i seguenti campi nel messaggio inviato tramite una richiesta http: • campi necessari per l'operazione di create/update (API ESB) • la codifica dell'elemento GQM+S contenuto nel messaggio inviato dalla fase. • la codifica da utilizzare nella memorizzazione dell'oggetto nello storage condiviso. Le codifiche supportate sono: • Base64 • JSON • Custom
F3	Gestione funzioni di traduzione di codifica	Il sistema permette all'Amministratore del sistema di effettuare operazioni di CRUD di funzioni di traduzione di codifica, definite attraverso il linguaggio javascript. Il sistema utilizza tali funzioni per nelle operazioni CRUD degli elementi GQM+S richiesti o inviati dalle fasi allo storage condiviso.

Modello dei casi d’uso

- Diagramma dei casi d’uso



Features	Caso d’uso
F1, F2	CRUD elementi GQM+S codificati
F3	Gestisci funzioni di codifica

- Breve descrizione dei casi d’uso
- Documentazione gruppo bus

Specifica caso d’uso<CRUD elementi GQM+S codificati>(1)

- Flusso base<read>

1. Il caso d'uso si attiva quando una Fase invia al sistema una richiesta http di tipo POST contenente nel body un messaggio, che specifica l’oggetto da recuperare e la codifica con cui si vuole ottenere l’elemento GQM+S («encode»).
2. Il sistema parsa la richiesta ricevuta, e recupera l'indirizzo del Sistema di storage.
3. Il sistema invia al Sistema di storage la richiesta dell'elemento/i GQM+S tramite un http POST.
4. Il Sistema di storage recupera e inoltra al sistema gli elementi GQM+S richiesti.
5. Il sistema accede al campo «payloadEncode» dell'iesimo elemento GQM+S e seleziona il codificatore da usare.
6. Il sistema cambia la codifica dell’iesimo elemento GQM+S.

   Il sistema ripete i passi 5 e 6 per ogni elemento GQM+S inviato dal Sistema di storage.
7. Il sistema costruisce il messaggio di risposta e lo invia alla fase.
8. La fase riceve l'elemento/gli elementi GMQ+S nella codifica richiesta.

BUS API(1)

- Create/update


```
{
    "tag":"level3Direct",
    "content": [
        "phase0",
        "create",
        "{\"objIdLocalToPhase\":\"3\",\"typeObj\":\"Assumption\",\"instance\":\"3\", \"tags\":[], \"encode\":\"base64\"
            ,\"payloadEncode\":\"JSON\",\"payload\":{\"criticalValue\":\"A1\",\"description\":\"A1\",\"id\":1,\"title\":\"A1\"
            ,\"creationDate\":\"2017-01-25\",\"projectId\":10,\"status\":0}}"
        ],
        "originAdress":"abc",
        "resolvedAdress":"",
        "id":""
}
```


-


```
{
    "tag":"level3Direct",
    "content": [
        "phase0",
        "read",
        "{\"objIdLocalToPhase\":\"3\", \"typeObj\":\"Assumption\", \"instance\":\"3\", \"tags\":[], \"busVersion\":\"\",\"encode\"
            :\"base64\"}"
        ],
        "originAdress":"abc",
        "resolvedAdress":"",
        "id":""
}
```

BUS API(2)

- Create funzione di traduzione di codifica in javascript:
  - Funzione per convertire il payload dalla codifica AF1 ad una codifica AF2 (Custom)

```
{
    "tag":"level3Direct",
    "content": [
        "fromAF1ToAF2",
        "var transform=function(data){data=JSON.parse(data);  /*elaborate data*/ return JSON.stringify(data);}"
        ],
        "originAdress":"updateEncode",
        "resolvedAdress":"",
        "id":""
}
```

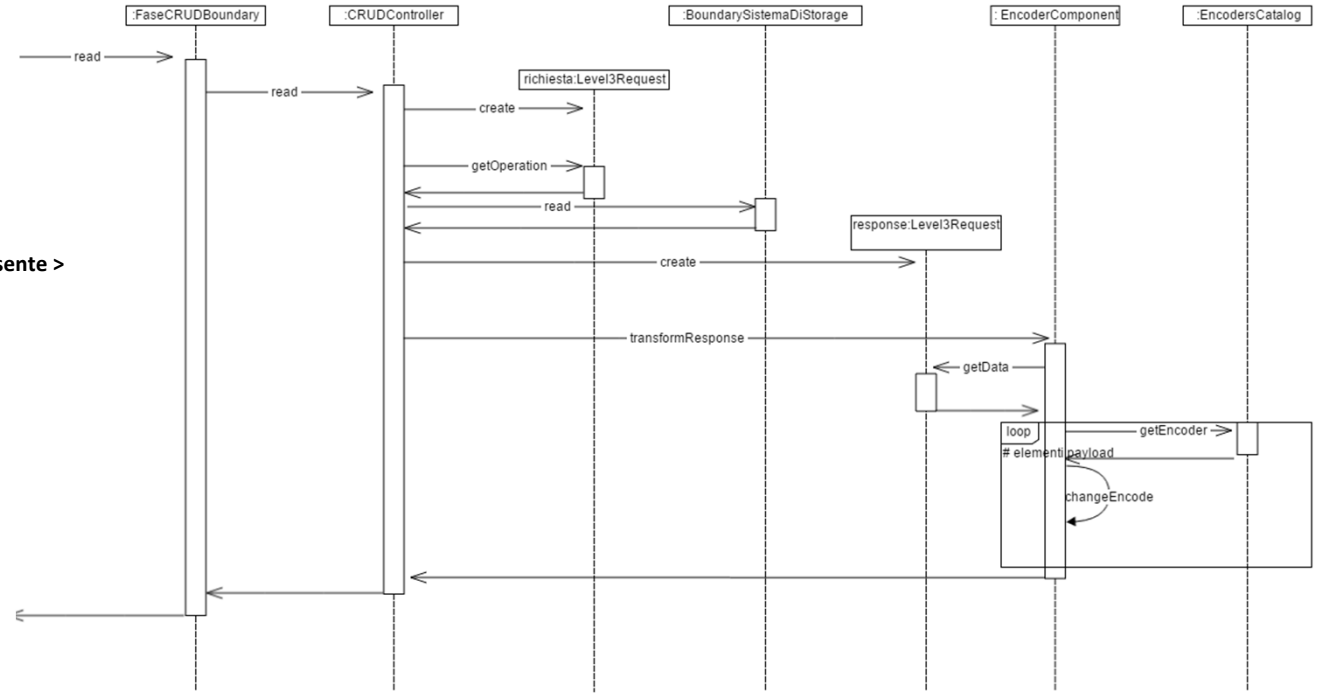
  - Esempio: Assumption
    - Fase1 instance: id|title Fase2.2
    - status->validationStat
    - Formato data

Realizzazione caso d'uso<CRUD elementi GQM+S codificati>(2)

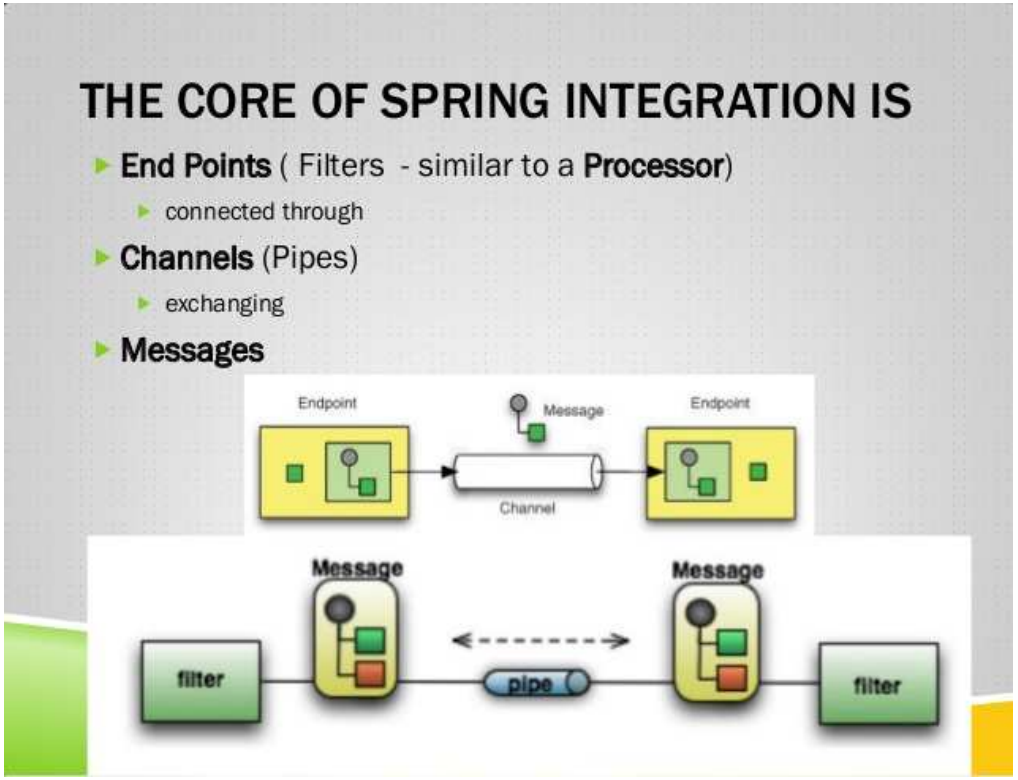
FLUSSO BASE READ

Flussi alternativi:

- **<Flusso alternativo Create>**
- **<Flusso alternativo Update>**
- **<Flusso alternativo Delete>**
- **<Flusso codificatore non presente >**
- **<Flusso errore codifica>**
- **<Flusso in old version>**
- **<Flusso out old version>**
- **<Flusso notificaion>**



Tecnologie utilizzate

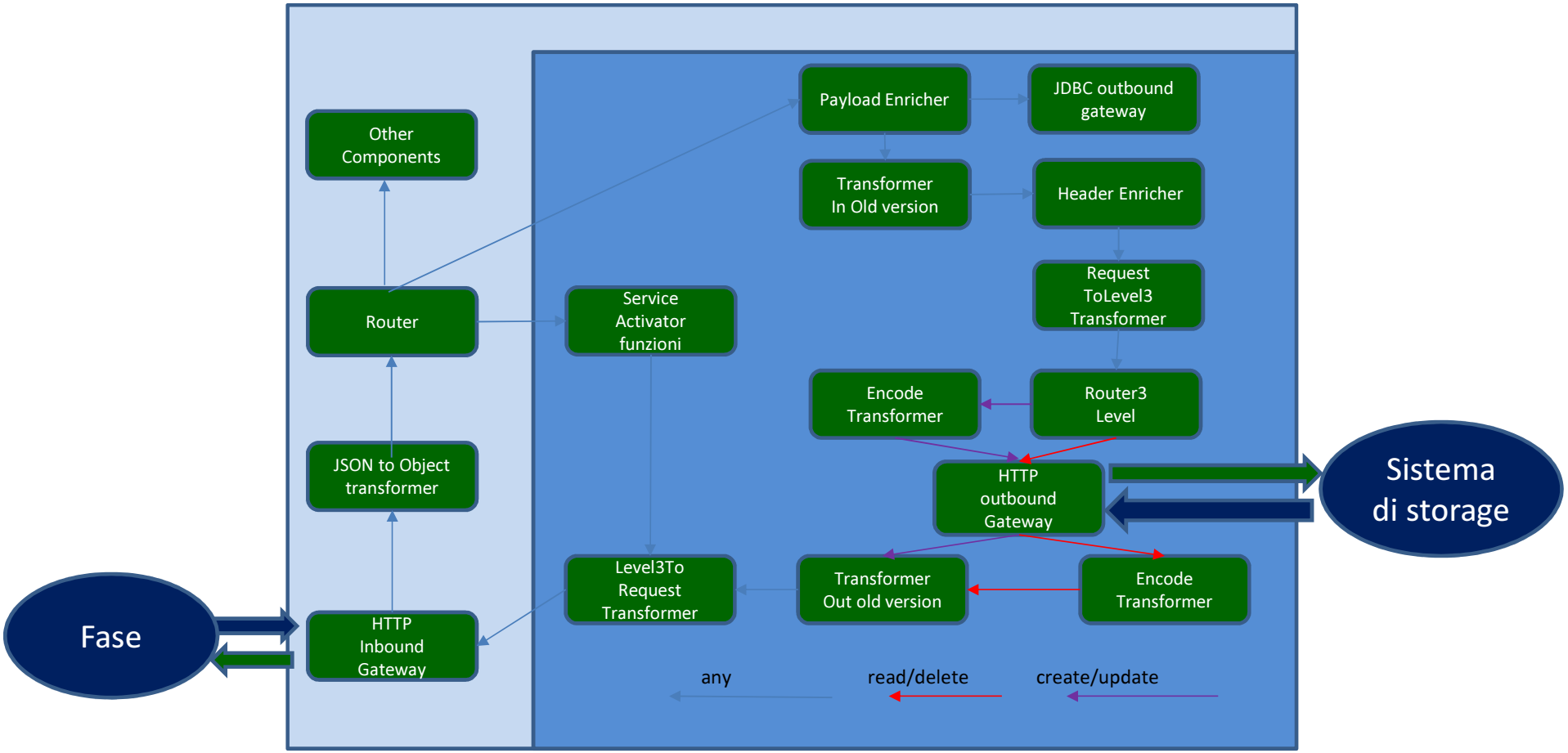


- Endpoint utilizzati
  - Service activator
  - Message enricher
  - Router
  - Transformer
  - Gateway

Diapositiva 14

M1 message: unità di informazione che può essere passata tra dierenti componenti;
endpoint: sono le componenti che si occupano dell'intervento sui messaggi, si va dal semplice routing al più complesso splitting ad operazioni di aggregazione;
channel: rappresentano la connessione tra endpoint.
Marco; 05/02/2017

Architettura ESB



Implementazione(1)

- Configurazione XML

```
<bean id="enricherService" class="it.enricher.EnricherService" />
<bean id="requestToLevel3Transformer" class="it.transformer.RequestToLevel3Transformer" />
<bean id="oldVersionTransformer" class="it.transformer.OldVersionTransformer" />
<int:router input-channel="toRouterLevel3" ref="Level3Router"
    method="select" default-output-channel="nonMatchesChannel" />
<int:transformer input-channel="level3WriteChannel"
    output-channel="toRequestLevel3DirectChannel" ref="encodeTransformer"
    method="transformRequest">
</int:transformer>
```

-

```
public class Level3Router {

    final static Logger logger = LoggerFactory.getLogger(Level3Router.class);

    public String select(Level3Request request) {
        if(request.getOperation().equals("create")||request.getOperation().equals("update"))
            return "level3WriteChannel";
        else
            if(request.getOperation().equals("read")||request.getOperation().equals("delete"))
                return "toRequestLevel3DirectChannelToTransform";
            else
                    return "toRequestLevel3DirectChannel";
    }
}
```

Implementazione(2):Invocazione funzione di traduzione

```
String javascriptTransformer= transformerDAO.getTransformer("from"+payloadEncode+"To"+encode);

if(javascriptTransformer!=null){
    ScriptEngine engine = new ScriptEngineManager().getEngineByName("nashorn");
    Object result = null;
    Invocable invocable = (Invocable) engine;
    try {
        String transformer = new String(Base64.getDecoder().decode((String)javascriptTransformer),
        engine.eval(transformer);
        result = invocable.invokeFunction("transform", data);
        if(result==null)
            throw new EncodeException("Javascript transformer exception: check javascript code");
        else
            return result.toString();
    } catch (ScriptException e) {
        throw new EncodeException("Javascript transformer exception: check javascript code");
    } catch (NoSuchMethodException e) {
        throw new EncodeException("Javascript transformer exception: check javascript code");
    } catch (UnsupportedEncodingException e) {
        throw new EncodeException("Base64 encoding error");
    }
}
else
    throw new EncodeException("Transformer not found");
```

Encode Transformer

Esempio

- READ JSON


```
{
  "tag": "level3Direct",
  "content": [
    "[{\"busVersion\":\"1\",\"instance\":\"10|A2\",\"err\":\"0\",\"payload\":{\"criticalValue\":\"A2\",\"description\":\"A2\",\"id\":10
        ,\"creationDate\":\"2017-02-05\",\"title\":\"A2\",\"projectId\":1,\"status\":0},\"typeObj\":\"Assumption\",\"payloadEncode\"
        :\"JSON\",\"tags\":[{\"value\":\"null\",\"key\":\"AssumptionTAG\"}]}]"
  ],
  "resolvedAdress": "",
  "originAdress": "",
  "id": null
}
```


-


```
{
  "tag": "level3Direct",
  "content": [
    "[{\"busVersion\":\"1\",\"instance\":\"10|A2\",\"err\":\"0\",\"payload\"
        :\"eyJjcml0aWNhbFZhbHVlIjoiQTIiLCJkZXNjcmlwdGlvbiI6IkEyIiwiaWQiOjEwLCJjcmVhdGlvbkRhdGUiOiIyMDE3LTAyLTA1IiwidGl0bGUiOiJBMiIsInByb2p
        lY3RJZCI6MSwic3RhdHVzIjowfQ==\",\"typeObj\":\"Assumption\",\"payloadEncode\":\"base64\",\"tags\":[{\"value\":\"null\",\"key\"
        :\"AssumptionTAG\"}]}]"
  ],
  "resolvedAdress": "",
  "originAdress": "",
  "id": null
}
```

Conclusioni

- Deploy
  - Bus5→ http://160.80.1.212/Bus5/inboundChannel.html
  - Sistema di storage→http://160.80.1.212/Integrazione5/level3Direct/storage/
- Test integrazione fase 0,1,2.1 e 2.2
  - Assumption, ContextFactor, OrganizationalGoal, notification
  - Jackson decoder fa
  - Fase 2.1 bug fix
- Sviluppo estensior

  - Programmazione di integrazione sostituita da configurazione
  - Supporto al cambiamento