



Metamorfosi di un sistema informatico

Un approccio pragmatico e sostenibile alla
modernizzazione di un sistema realizzato
con tecnologia Oracle

Roma, 4 Aprile 2019

Speaker

Andrea Ferroni

Solution Developer Senior
Tech Lead, Analyst, Developer

ITOUG board member

Email: andrea.ferroni@itoug.it



ORACLE FORMS

Una strategia di
migrazione, ma
non solo

Metamorfosi

Il bruco

Il sistema esistente



La crisalide

Evoluzione del
back-end



La farfalla

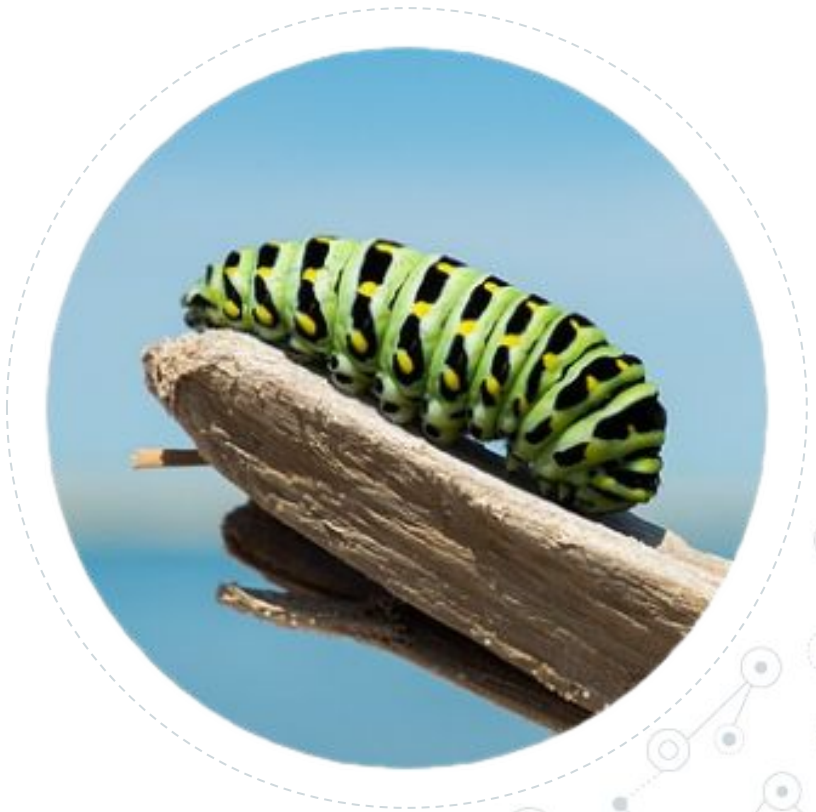
Un' alternativa
per la UI



1.

Il bruco

Il sistema
esistente





Vasto

Ecosistema
applicativo di
ampie
dimensione

3.000

FORMS

5.000

Tabelle

1.300

Package



Eterogeneo

Tecnologie diverse
coesistono

UI

ORACLE FORMS

BL

EXE PRO*C

PACKAGE PL\SQL

Interattiva

Applicazioni Table Based e data entry

Batch

Processi elaborativi massivi automatici non presidiati

Operatività

Due modalità



Change

Oracle Forms 6.0

Oracle Forms 11g

Vincoli

Riutilizzare EXE core

Riutilizzare BL (Package)

Supporto processi (BATCH)

2.

La crisalide

La metamorfosi
ha inizio



A decorative background consisting of a network of interconnected nodes and lines, rendered in light gray. The nodes are represented by small circles, some of which are larger and have a double-circle effect. The lines are thin and connect the nodes in a complex, web-like pattern. The overall aesthetic is clean and technical.

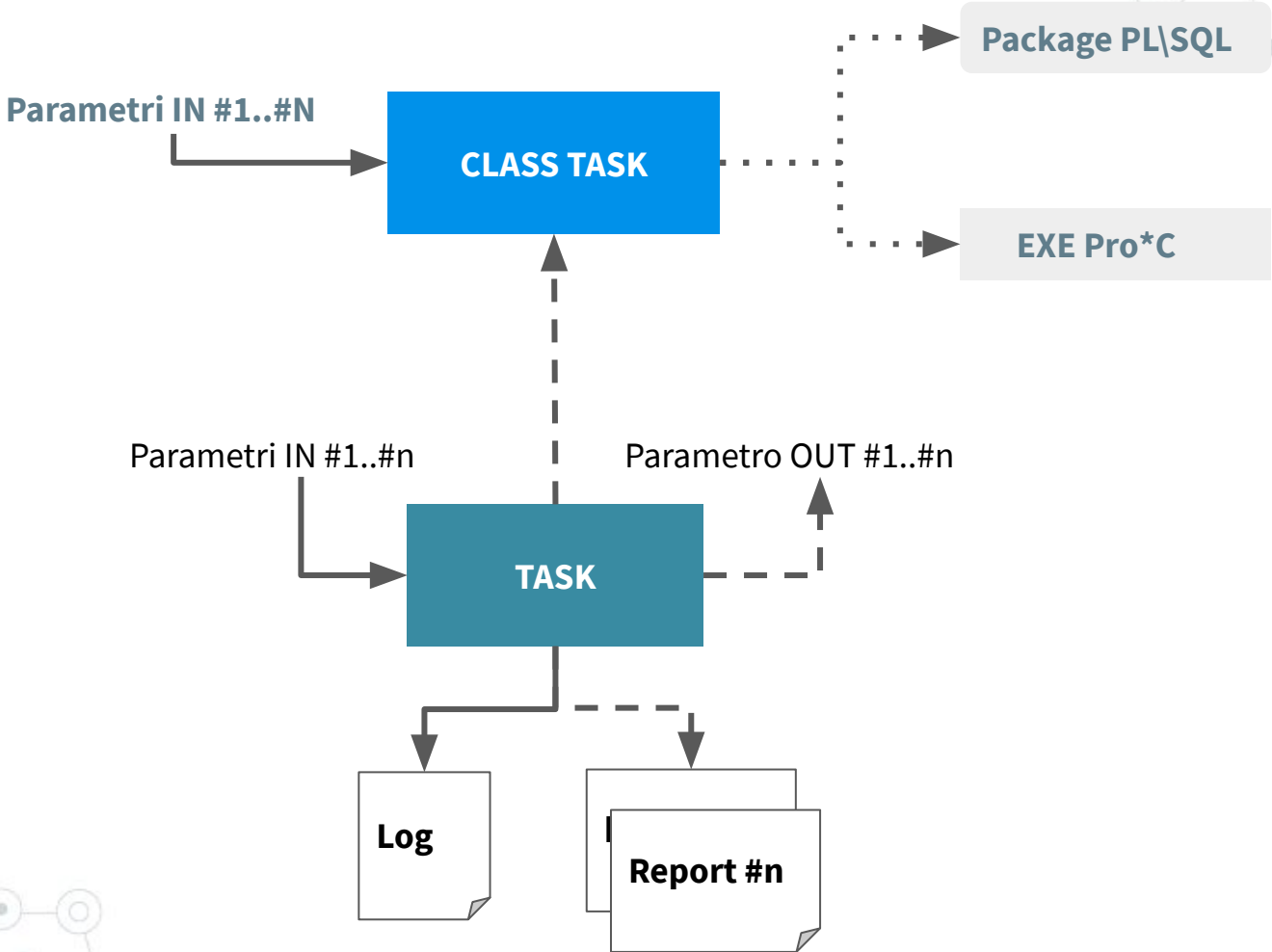
SCHEDULER

ORACLE SCHEDULER 11g

SDK

FRAMEWORK PL\SQL

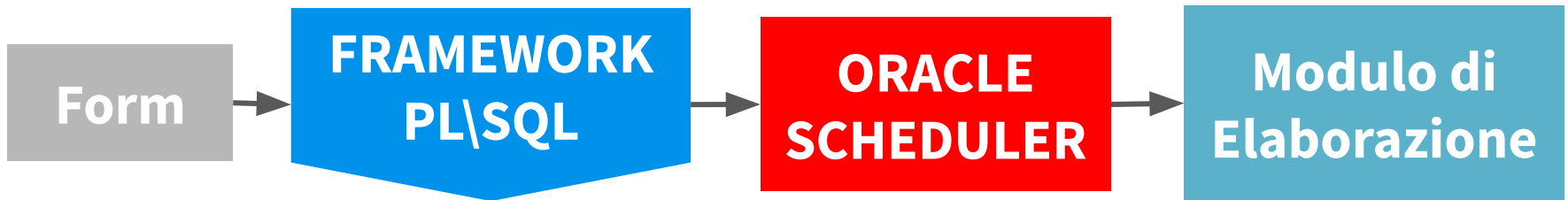
TASK



TASK CALL



```
idTask :=FRAMEWORK.TaskAdd('<CLASS_TASK>');  
FRAMEWORK.TaskRun(idTask);
```



```
PROCEDURE FRAMEWORK.TaskRun(task IN tTask) IS  
    DBMS_SCHEDULER.CREATE_JOB('BEGIN <PACKAGE.PROCEDURE>(' || task.id ||')  
END');  
END;
```

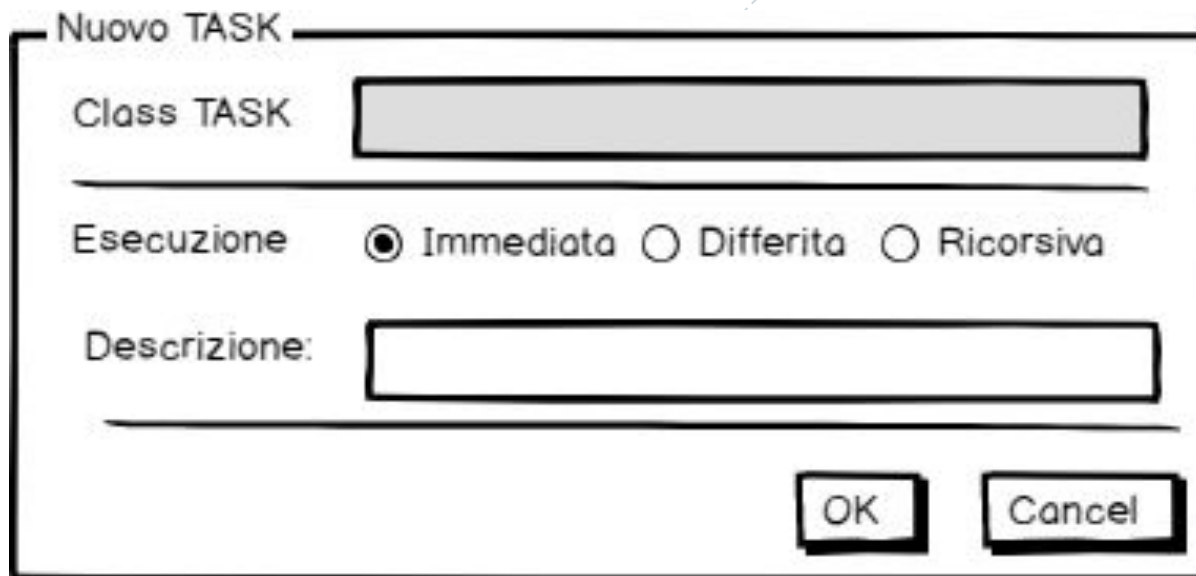
Implement TASK

Procedura PL\SQL che implementa un TASK

```
CREATE OR REPLACE PACKAGE ORDER_PCK AS PROCEDURE
    PRINT(taskId IN VARCHAR2);
END;
/
CREATE OR REPLACE PACKAGE BODY ORDER_PCK AS PROCEDURE PRINT(taskId IN VARCHAR2) IS
    log          FRAMEWORK.tLog;
    file         FRAMEWORK.tFile;
    paramList    FRAMEWORK.tParam;
BEGIN
    FRAMEWORK.NOTIFY ('OrderPrint BEGIN');
    log := FRAMEWORK.LOG_OPEN(taskId, 'ORDER_PCK_' || taskId || '.LOG');
    paramList := FRAMEWORK.PARAM_LIST_LOAD(taskId, log);
    FRAMEWORK.PARAM_PRINT(taskId, log);
    file:= FRAMEWORK.FILE_NEW(taskId, log);
    --Print Order param.OrderID
    FRAMEWORK.FILE_PRINT(taskId, 'OrderId:' || paramList('ORDERID') , log);
    FRAMEWORK.FILE_CLOSE(file,log);
    FRAMEWORK.LOG_CLOSE(log);
    FRAMEWORK.NOTIFY ('OrderPrint END');
END;
/
```

FORM Dialog

Modalità di esecuzione TASK



Nuovo TASK

Class TASK

Esecuzione Immediata Differita Ricorsiva

Descrizione:

FORM Monitor

Monitoraggio esecuzione TASK

Lista TASK

Class TASK

ID TASK	Class TASK	Descrizione	Stato	Inizio	Termine			
						Parameters...	Stop	Files...
						Parameters...	Stop	Files...
						Parameters...	Stop	Files...

XFORM

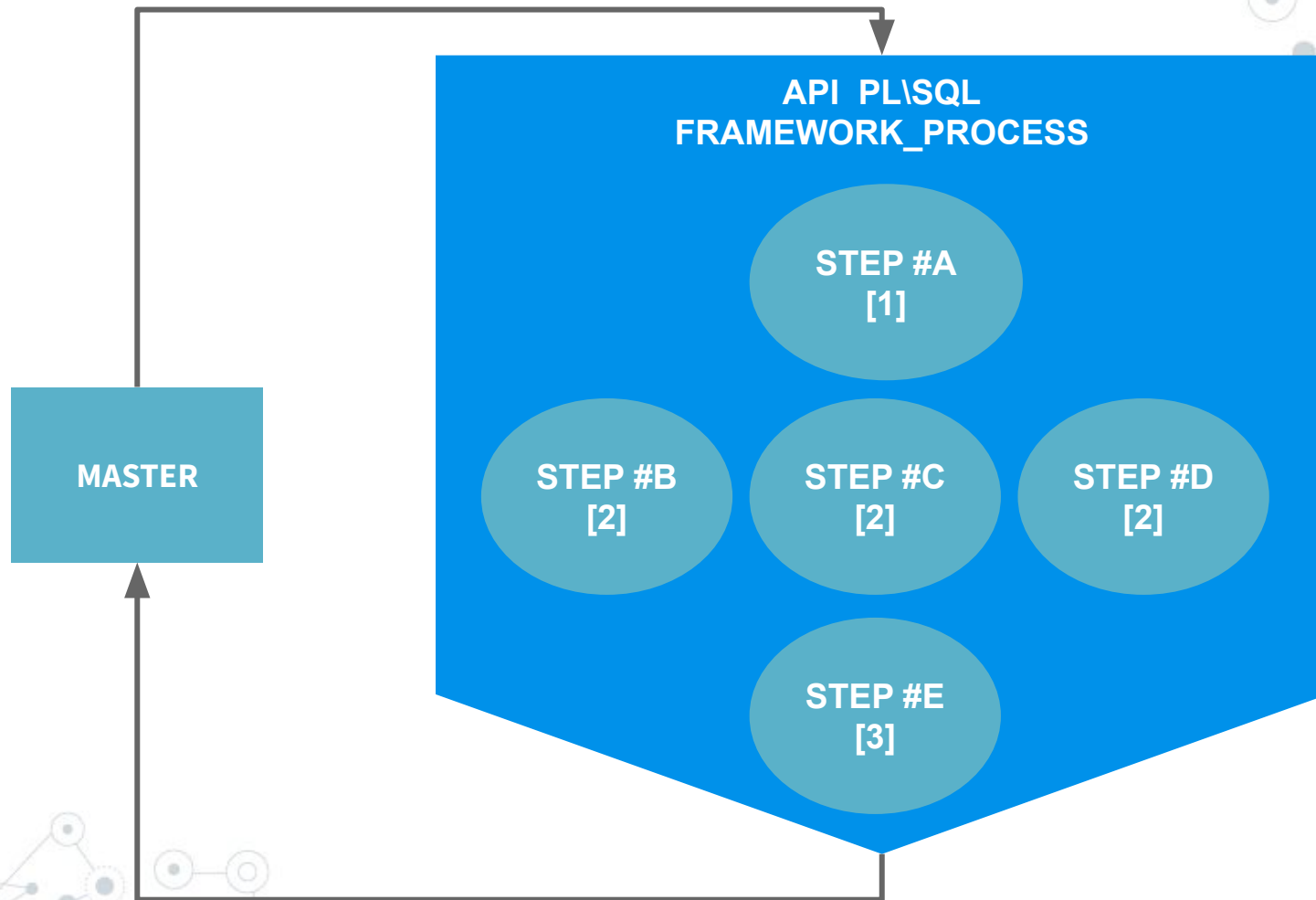
XFORM

Class TASK

Descrizione	Valore
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Esegui TASK

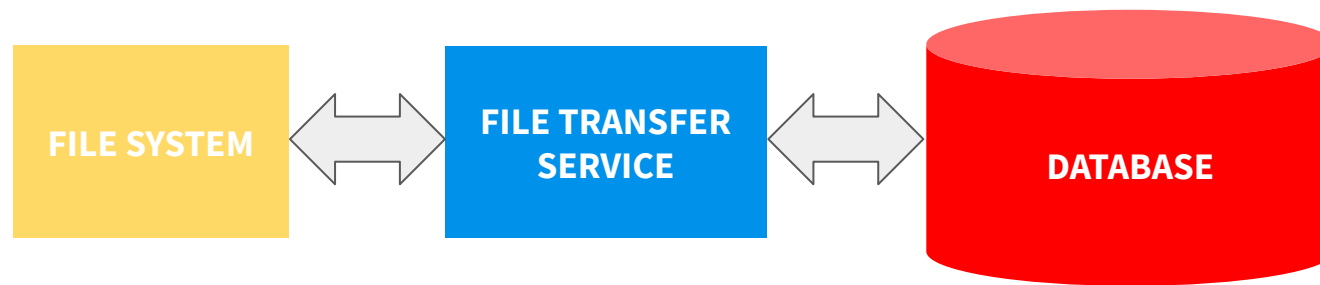
TASK CHAIN



TASK-BOT

- © **TASK** schedulato per essere eseguito in modo ripetitivo con una certa periodicità
- © Abilita all'automazione di processi di business

File Transfer Service



- © **FTS Servizio Windows C#**
- © **Trasporto di FILE**
- © **TASK-BOT** per l'acquisizione o la generazione automatica di **FILE**



PRO

- ◎ Storicizzazione elaborazioni
- ◎ **BL** incapsulata in moduli componibili
- ◎ Disaccoppiamento **UI/BL**
- ◎ Componenti **UI** centralizzate e minimizzate
- ◎ Censimento dei moduli di **BL**
- ◎ Focus funzionale

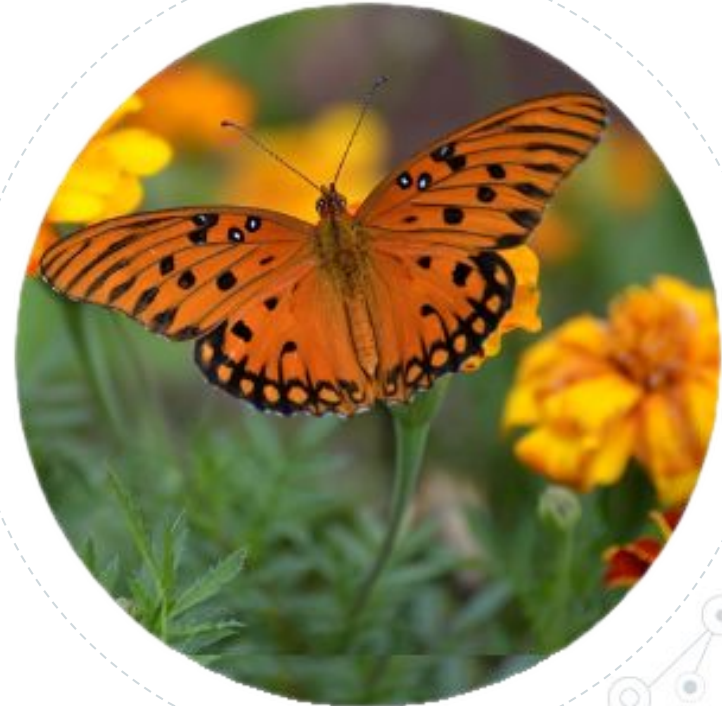


CONTRO

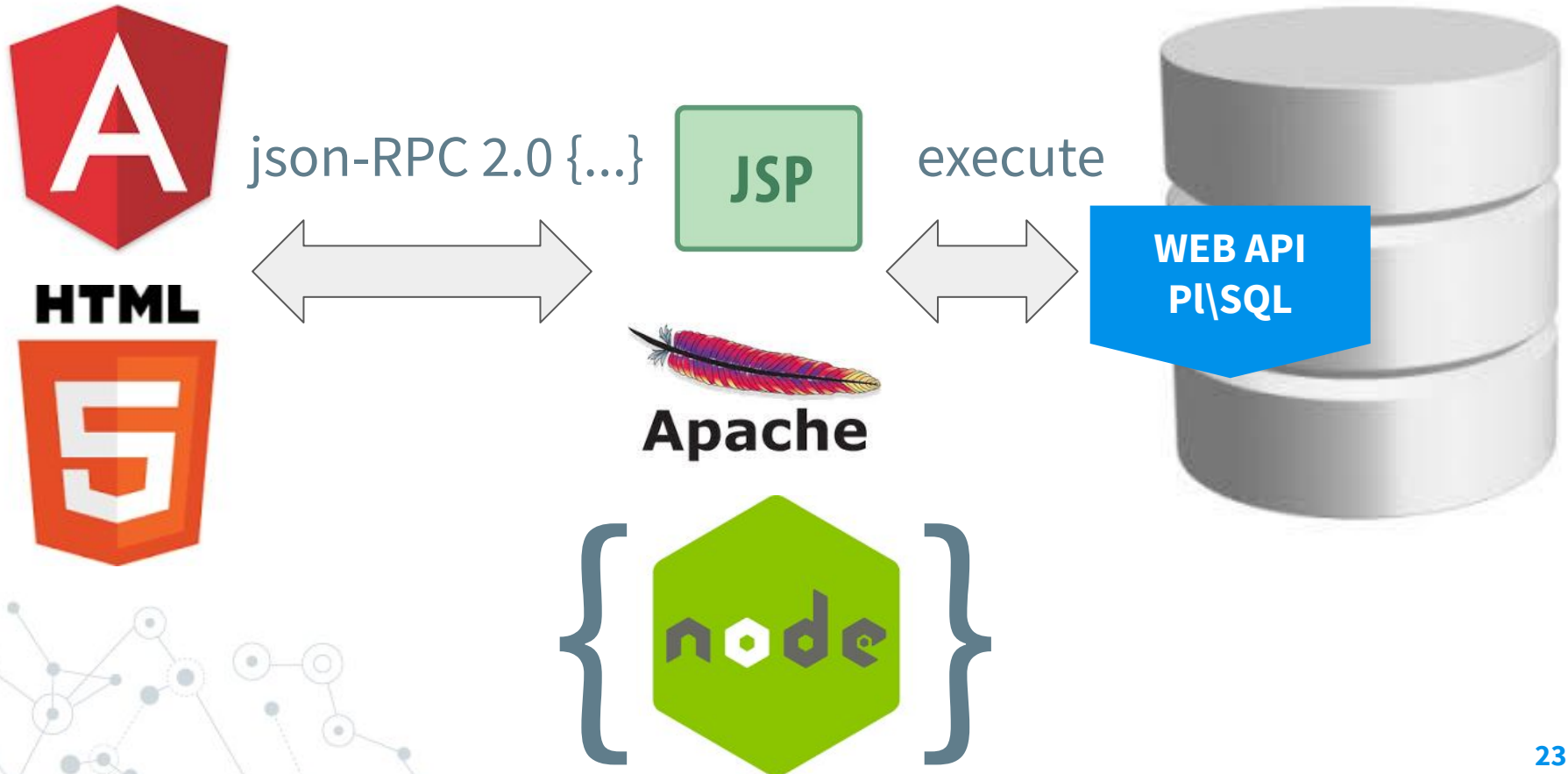
- ◎ Centralità del FRAMEWORK
- ◎ Occupazione di spazio DB

3. **La farfalla**

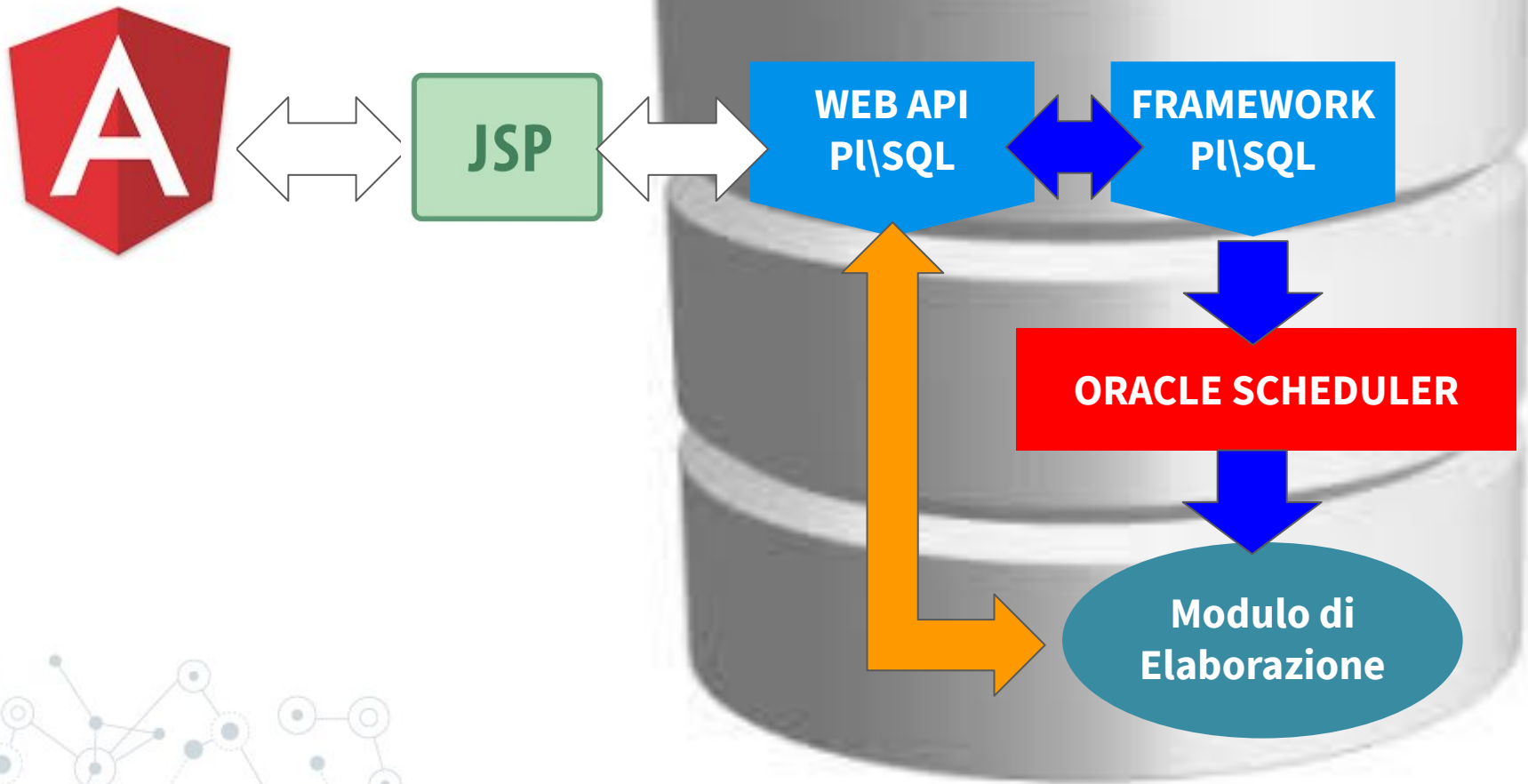
La metamorfosi
compiuta



Architettura



Interactive & Batch





4.

Futuro

Ulteriori possibili evoluzioni

Sistema ad eventi

- ◎ Un sistema reattivo composto da numerosi **TASK-BOT**
- ◎ **TASK** che cooperano tra loro attraverso una coda di messaggi
- ◎ **PRO**
 - Semplice da disegnare
 - Vicino alla semantica dell'analisi:
 - ◎ “quando arriva un ordine viene automaticamente verificato, stampato”
 - ◎ “ogni volta che viene generato un documento viene notificato al sistema integrato di contabilità”
- ◎ **CONTRO**
 - Delicato
 - Deploy complicato



5. **Riflessioni**

Qualche considerazione

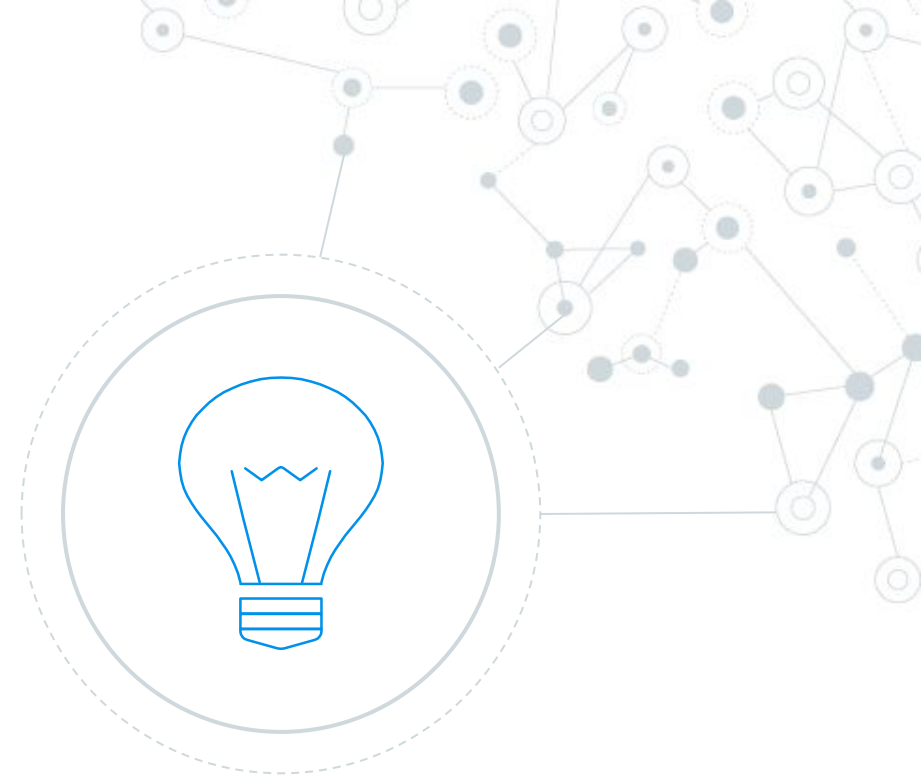
Longevità

La longevità di un sistema dipende più dell'architettura che dalla tecnologia



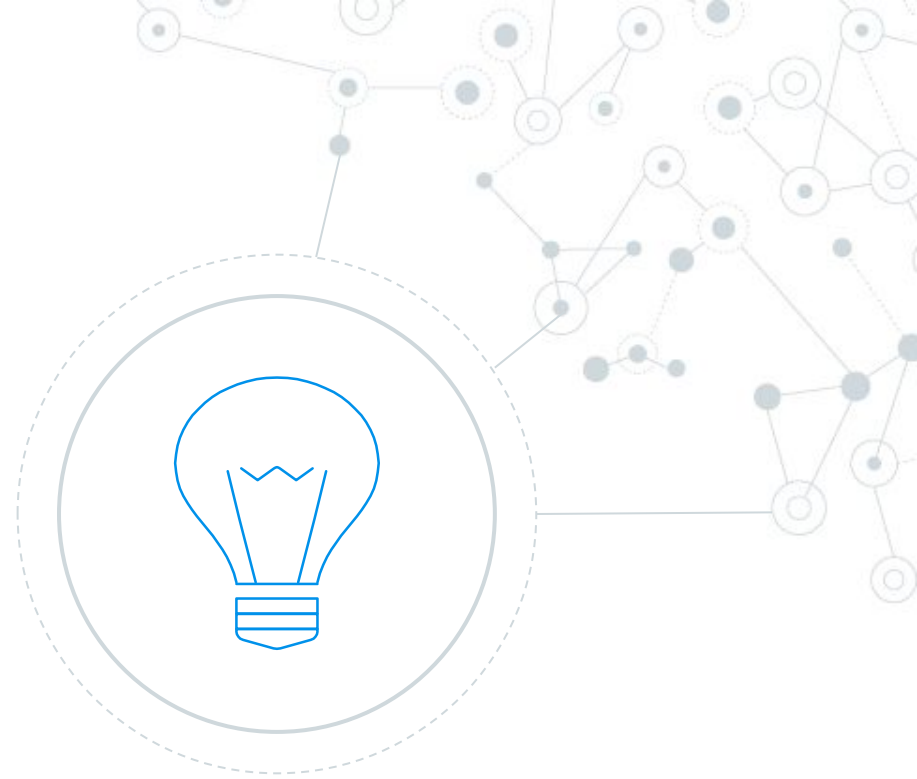
Architettura

L'architettura di sistema
può essere abilitante a
livello applicativo



UI

La UI invecchia più velocemente della BL perciò è fondamentale disaccoppiarle.





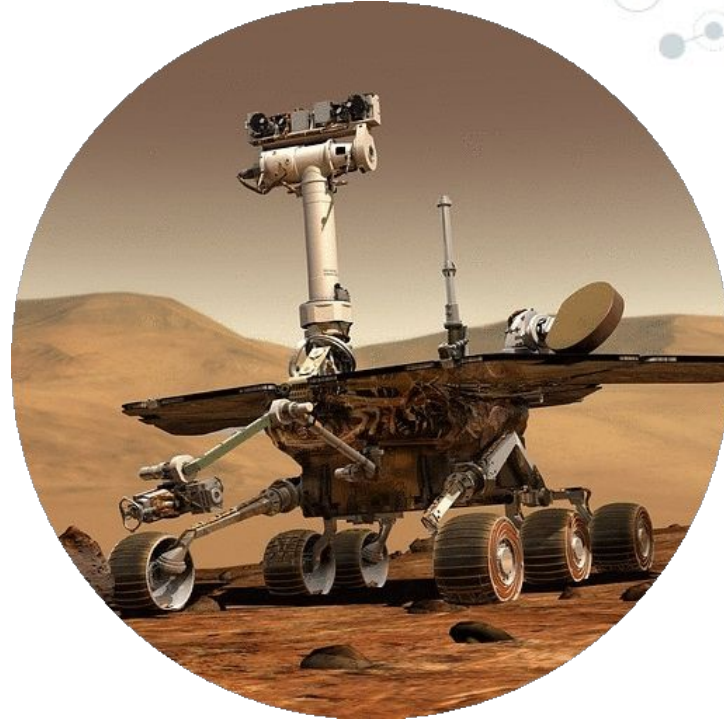
Grazie!

Innovazione frugale: l'arte di escogitare soluzioni ingegnose usando strumenti semplici.



Domande

Quanto è longevo
il tuo sistema ?



andrea.ferroni@itoug.it

Credits

- © Template presentazione Cordelia da [SlidesCarnival](https://slidescarnival.com)
- © Foto da <https://pixabay.com>