("WNoDeS in OpenStack")

Davide Salomoni
INFN CNAF
28/5/2013

Wack: definitions

wack

I

n.

1 (colloq) pazzo **m.**, matto **m.**, spasso **m.**

2 (Br,region) amico **m.**

II

agg.

1 passo, matto, spassoso.

2 (Am,colloq) terribile, orribile.

III

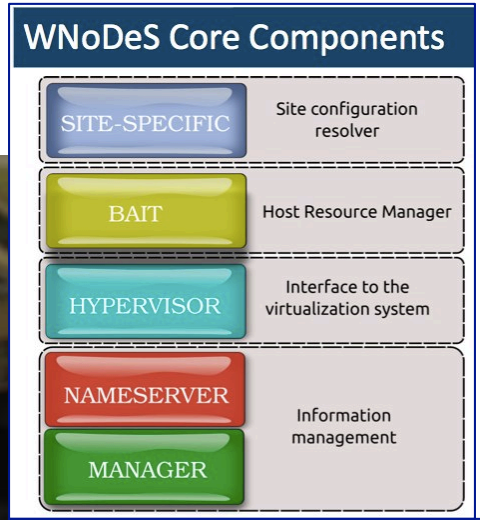
v.tr.

1 colpire rumorosamente, battere rumorosamente.

2 (colloq) (to thrash) battere, percuotere, picchiare; (to defeat) sconfiggere, battere.

3 (colloq) (to divide into shares) spartire, dividere.

4 (Am,sl) (to kill) freddare.



openstack™
CLOUD SOFTWARE

Stable release	Grizzly (2013.1.1) / May 10, 2013; 17 days ago
Written in	Python
Type	Cloud computing
License	Apache License 2.0
Website	openstack.org

Le componenti

- OpenStack
  - Cloud computing IaaS framework
- WNoDeS
  - Framework per l'integrazione di workload di tipo scientifico in modalità local, Grid e Cloud. Due caratteristiche importanti:
    - Mixed mode (esecuzione contemporanea di workload su bare metal e su VM)
    - Resource provisioning attraverso un batch system (es. LSF, PBS/Torque, SLURM)
- Perché integrazione?
  - Sostenibilità (di WNoDeS)
  - Supporto di use case attualmente non gestiti dall'uno o dall'altro prodotto
- Perché OpenStack?
  - Cf. presentazione generale su OpenStack. Niente comunque proibisce in linea di principio l'integrazione con altri stack Cloud.

Timeline di WNoDeS

- Attualmente in programma revisioni di codice, stabilizzazione del core
  - Focalizzandosi sulle caratteristiche peculiari di WNoDeS
- Ma non ha senso perseguire la scrittura di codice che duplichi funzioni di tipo ad es. “Cloud” già presenti altrove
  - Come gestione di OCCI server, implementazione di altre interfacce/API, Web-based UI

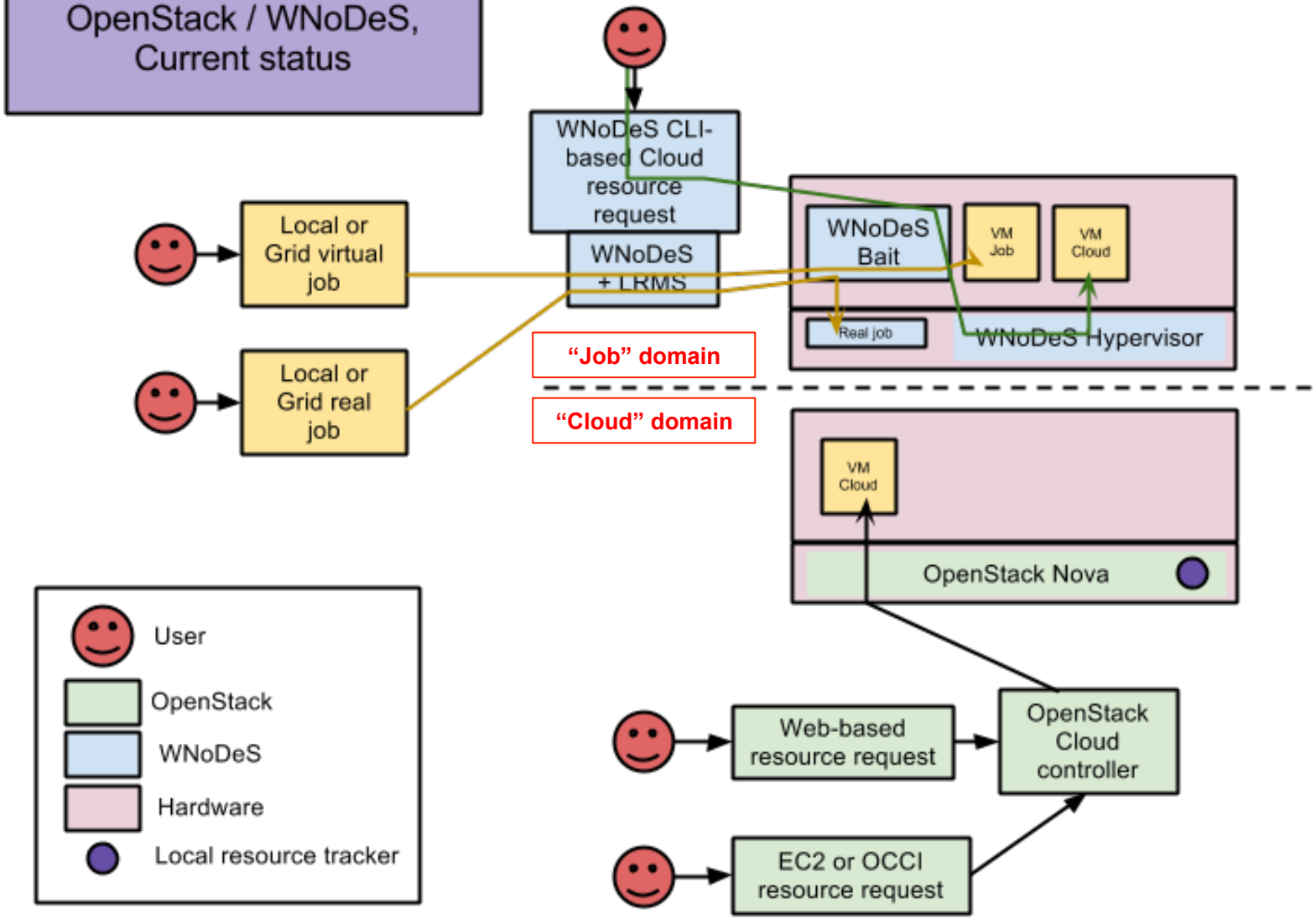
Use case

- Alcuni esempi d'uso non necessariamente coperti da stack "Cloud" presenti sul mercato:
  - Utilizzo di una farm di grandi dimensioni per gestire calcolo scientifico sia con job "reali" (senza virtualizzazione) sia con job che devono andare in esecuzione su VM.
    - I job possono essere locali oppure di tipo Grid.
    - Non deve essere *necessario* partizionare le risorse in (ad es.) risorse che eseguono job "reali" e risorse che eseguono job su VM.
  - Supporto di istanziazioni di tipo Cloud, ad esempio per:
    - VM per servizi
    - VM "personali" create via Web interface o via CLI/API
    - Pool di VM
    - Cloud storage di tipo object (S3, Swift) o di tipo volume (EBS, Cinder)
    - Non deve essere *necessario* partizionare le risorse in (ad es.) risorse di tipo "Cloud" e risorse di tipo "non Cloud".

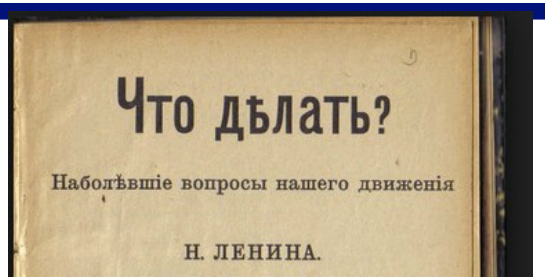
Analisi delle due componenti

- WNoDeS
  - Gestisce risorse (virtuali o no) di calcolo attraverso una stretta integrazione con un batch system, che funge da allocatore di risorse
    - Dimostrata scalabilità in produzione (al Tier-1 INFN) a diverse migliaia di VM in esecuzione contemporanea
    - Supporta istanziazioni “alla Cloud” attraverso una API OCCI-like
- OpenStack
  - Gestisce scenari di tipo Cloud per allocazione di VM, reti, storage; supporta monitoring, migrazione tra VM, snapshot, Web interface, API, networking, etc.
    - Da “vero framework Cloud”, non supporta il concetto di “accodamento” delle richieste di allocazione risorse

OpenStack / WNoDeS, Current status



Obiettivo

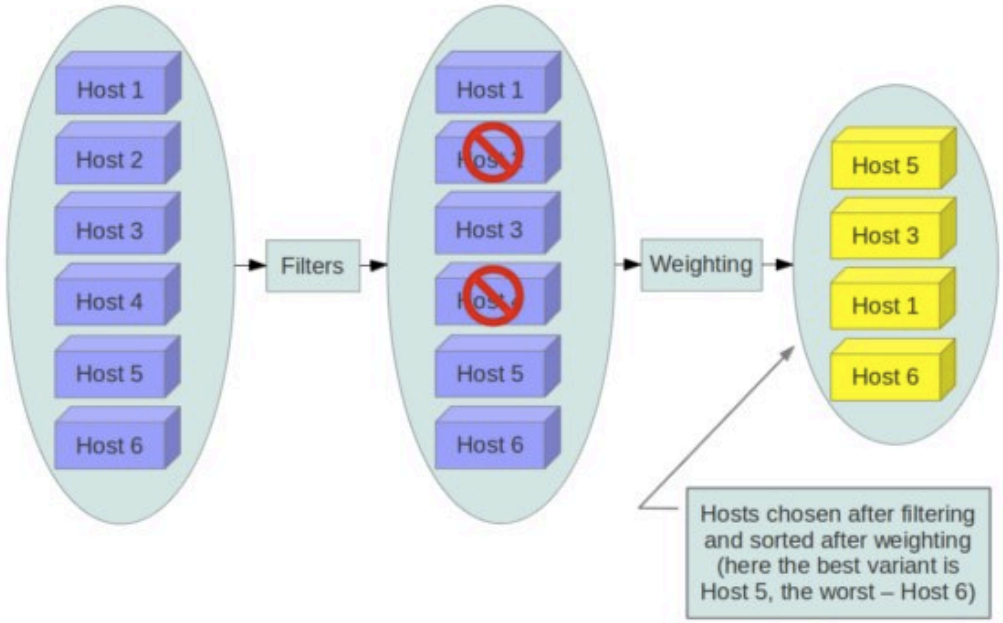


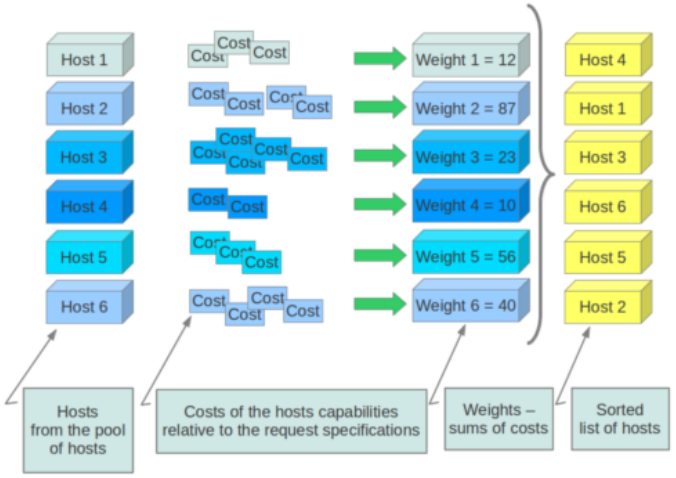
- Semplificazione di WNoDeS con riutilizzo della maggior parte possibile di componenti di OpenStack
  - Convergenza sulla gestione delle immagini e utilizzo di Glance
  - Eliminazione dello strato “custom” di virtualizzazione e utilizzo di Nova
  - Non verrà implementata un’interfaccia “WNoDeS-specific” di EC2 (o una revisione dell’attuale interfaccia OCCI-like)
- Mantenimento / espansione delle funzionalità **core** di WNoDeS per la gestione di una farm omogenea (non partizionata) i cui compute nodes abbiano installato sia WNoDeS sia OpenStack per la gestione contestuale di workload locali, Grid e Cloud.

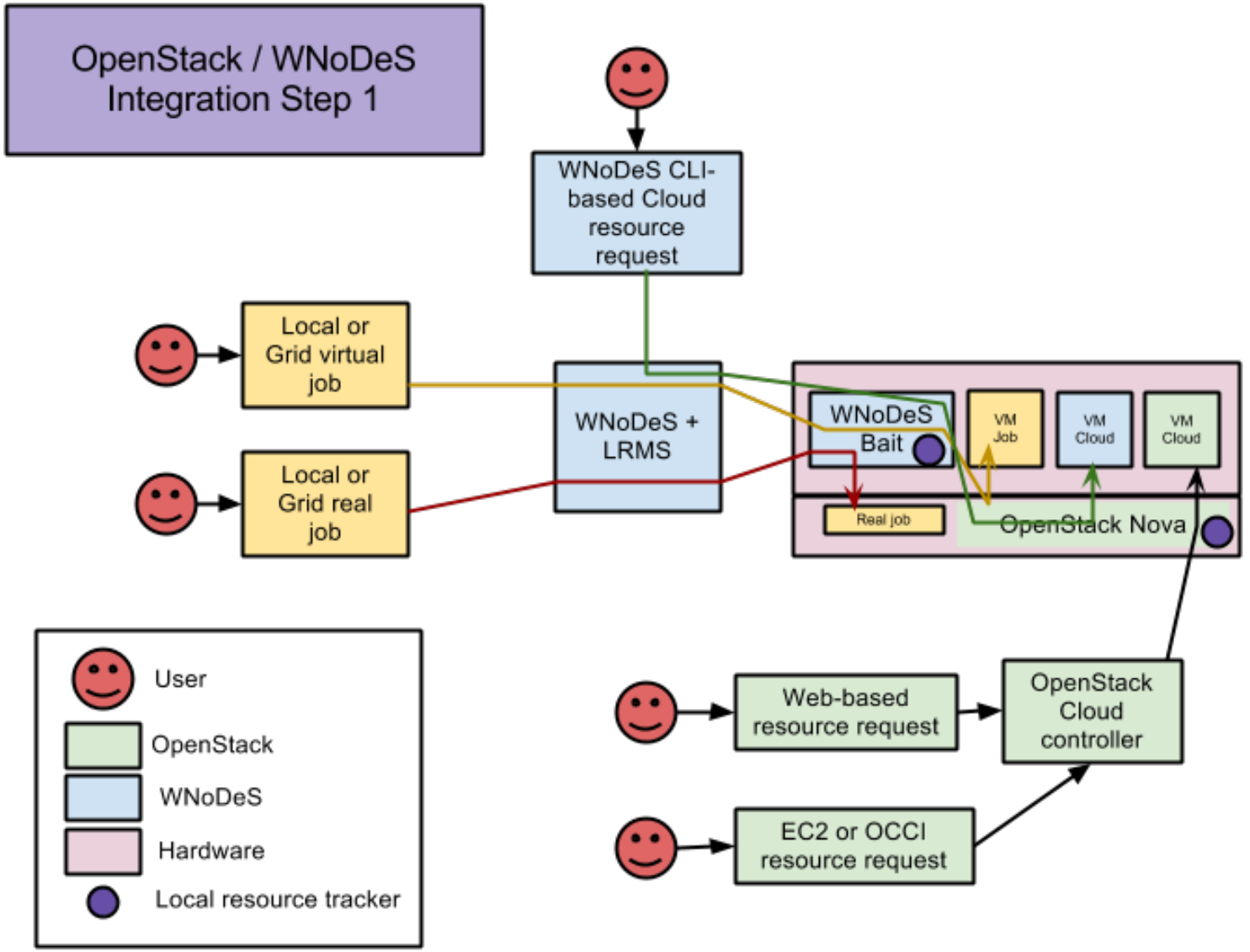
Lo scheduler di OpenStack

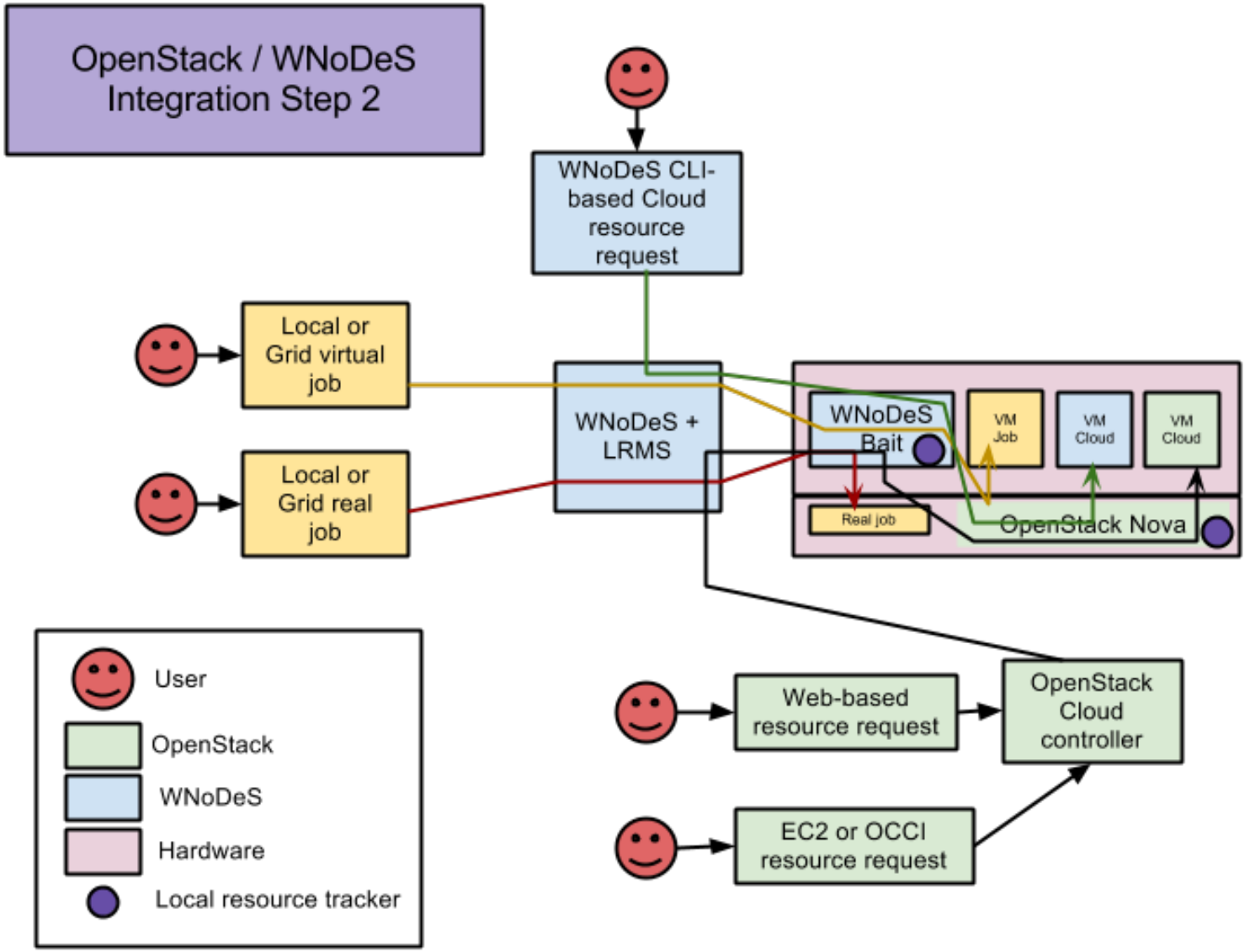
- Descritto in http://goo.gl/sLXCo
  - Qualche dettaglio in più su http://goo.gl/o4Swo
  - Punto importante: no risorse libere → no VM

```
All compute nodes (also known as hosts in terms of OpenStack) periodically publish
their status, resources available and hardware capabilities to nova-scheduler through
the queue. nova-scheduler then collects this data and uses it to make decisions when
a request comes in.
```





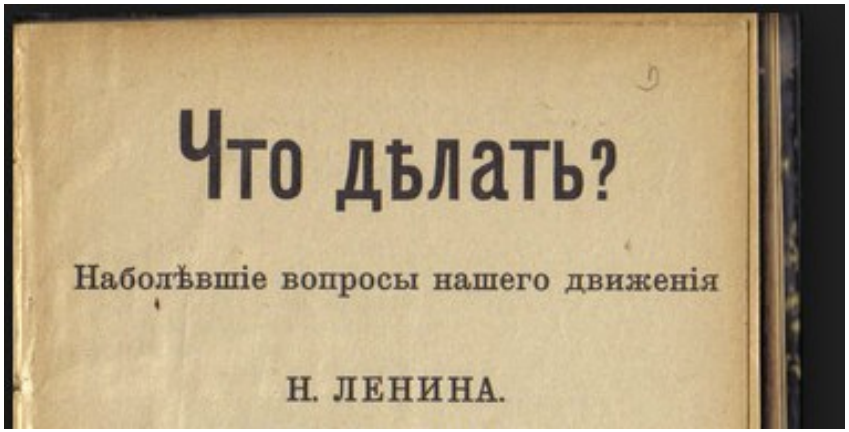

OpenStack / WNoDeS Integration Step 1
WNoDeS CLI-based Cloud resource request
Local or Grid virtual job
Local or Grid real job
WNoDeS + LRMS
WNoDeS Bait
VM Job
VM Cloud
VM Cloud
Real job
OpenStack Nova
Web-based resource request
EC2 or OCCI resource request
OpenStack Cloud controller
User
OpenStack
WNoDeS
Hardware
Local resource tracker


OpenStack / WNoDeS Integration Step 2
WNoDeS CLI-based Cloud resource request
Local or Grid virtual job
Local or Grid real job
WNoDeS + LRMS
WNoDeS Bait
VM Job
VM Cloud
VM Cloud
Real job
OpenStack Nova
User
OpenStack
WNoDeS
Hardware
Local resource tracker
Web-based resource request
EC2 or OCCI resource request
OpenStack Cloud controller

Cosa fare

- Fase 1:
  - Integrazione della gestione delle immagini attraverso Glance
  - Coesistenza tra WNoDeS e OpenStack sugli stessi compute node
    - Modifica del local resource tracker di OpenStack
    - OpenStack continua in questa fase a non gestire l'accodamento delle richieste di allocazione risorse
- Fase 2:
  - Estensione dello scheduler di OpenStack per consentire istanziazione (opzionale?) attraverso un batch system
- Partecipazione attiva all'OpenStack Consortium
- Con chi:
  - Tier-1 (integrazione VM, job "reali", Cloud)
  - Esperimenti (discussioni in corso con CMS per accesso integrato via Condor/EC2)
  - PRISMA

Addendum: What is to be done?



- In *What Is to Be Done?*, ~~Lenin~~ D. Salomoni argues that ~~the working class~~ INFN will not spontaneously ~~become political~~ adopt Cloud computing simply by fighting ~~economic~~ ideological battles with employers over ~~wages, working hours~~ superiority of the Cloud, superiority of the Grid, market trends, staff positions and the like. To convert ~~the working class~~ INFN to ~~Marxism~~ Cloud computing, ~~Lenin~~ D. Salomoni insists that ~~Marxists~~ INFN should form a ~~political party~~ strategic, technical vision, or "vanguard", of dedicated revolutionaries to spread ~~Marxist political~~ technical ideas about distributed computing integration among the workers.
  - Source: Wikipedia (adapted)