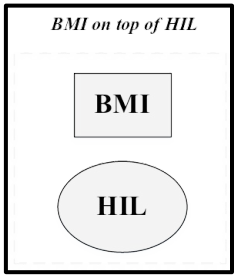
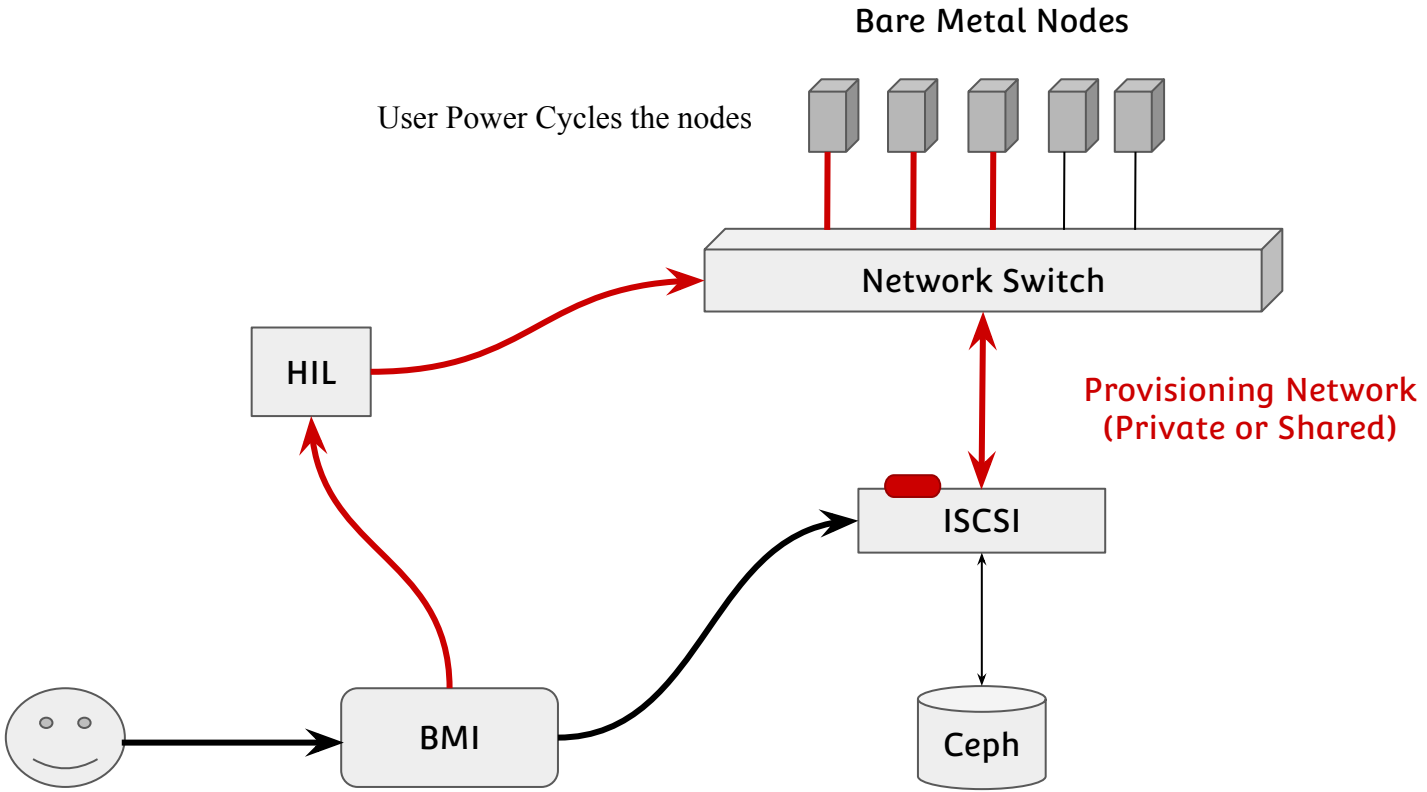
BMI Redesign

Current Design (Single Provisioning Network e.g. MOC Research Groups)





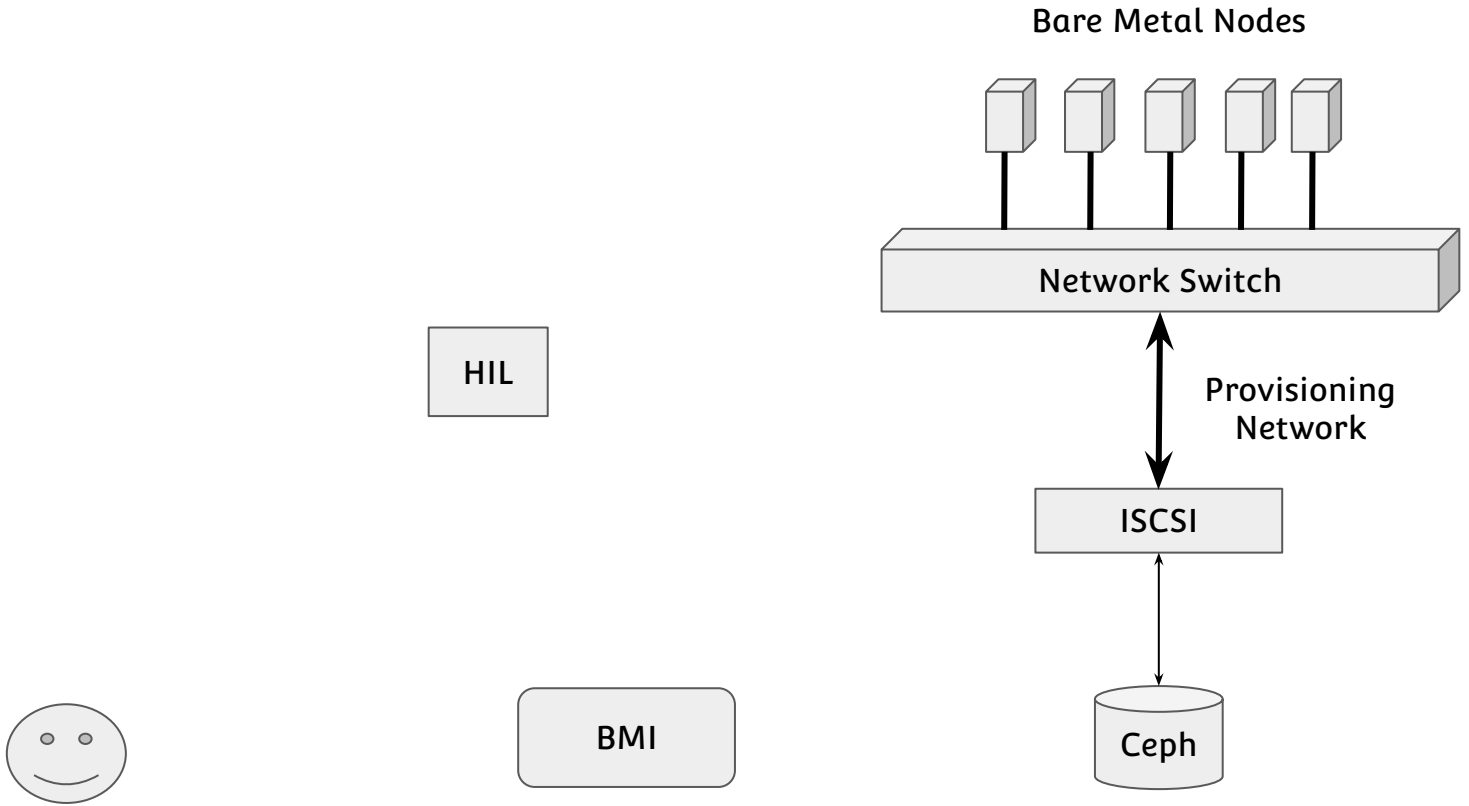
Deployment Scenarios and Use Cases

1. Flat Provisioning Network - Single Tenant Single Network *(Tenant owned BMI service)*
   - **Use case**
     - OpenNebula
     - SecCloud (Air Force)

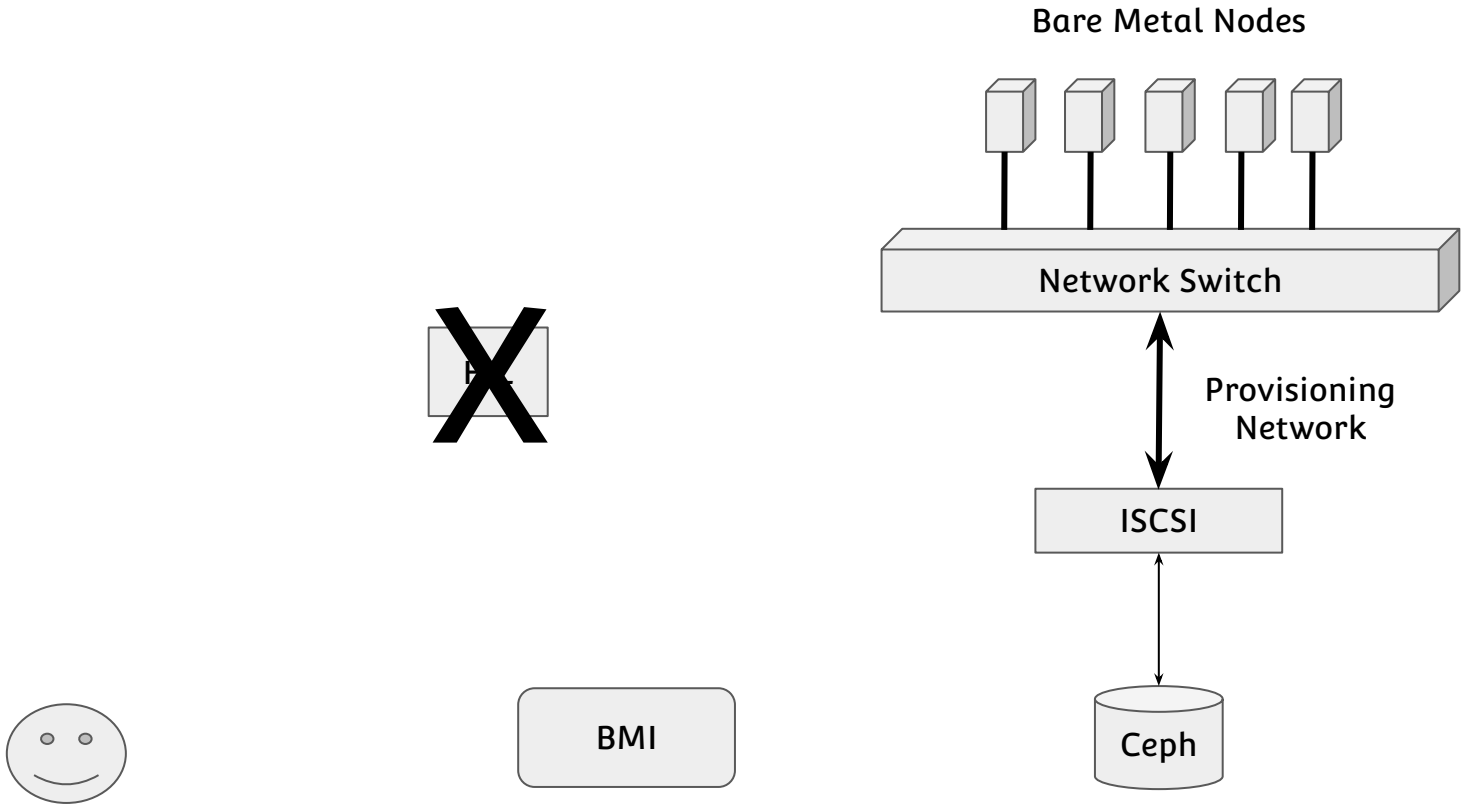
2. Shared Provisioning Network - Multiple Trusted Tenants Single Network *(Provisioning as a Service)*
   - **Use case**
     - MOC Research Groups

3. Private Provisioning Networks - Multiple UnTrusted Tenants Multiple Networks *(Provisioning-as-a-Service)*
   - **Use case**
     - SecCloud (Air Force)

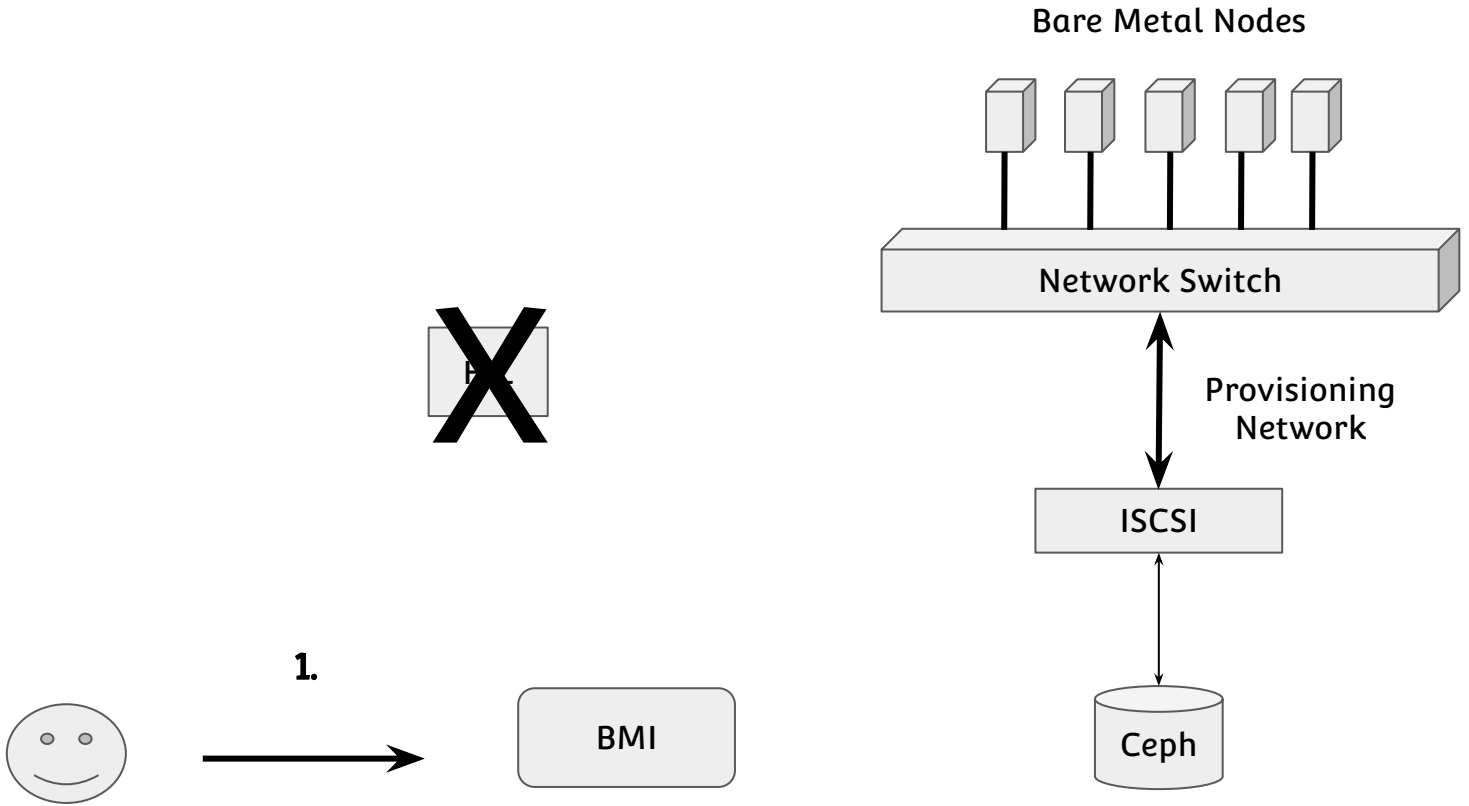
Proposed Design (Flat Provisioning Network - Single Tenant - e.g. OpenNebula)



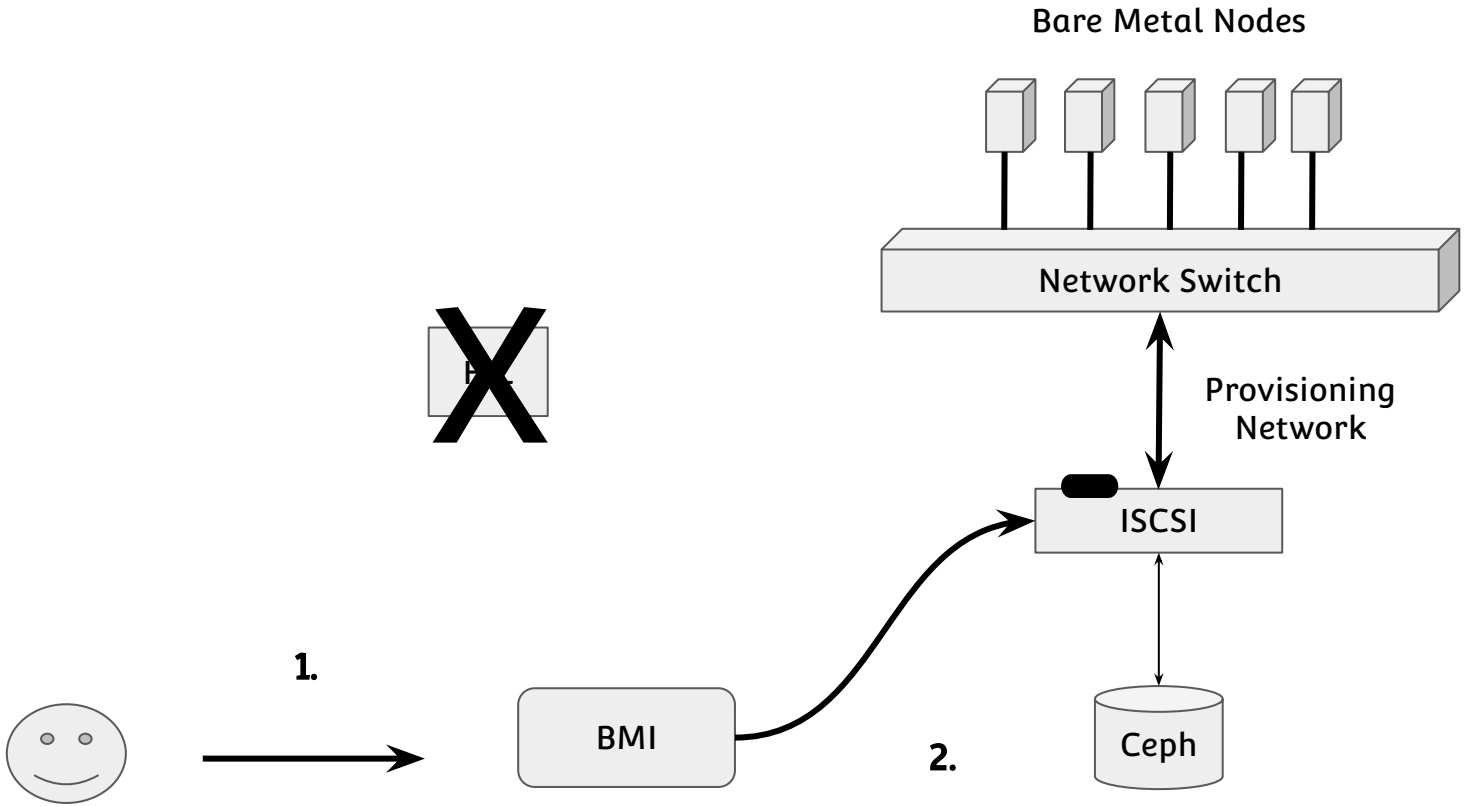
Proposed Design (Flat Provisioning Network - Single Tenant - e.g. OpenNebula)



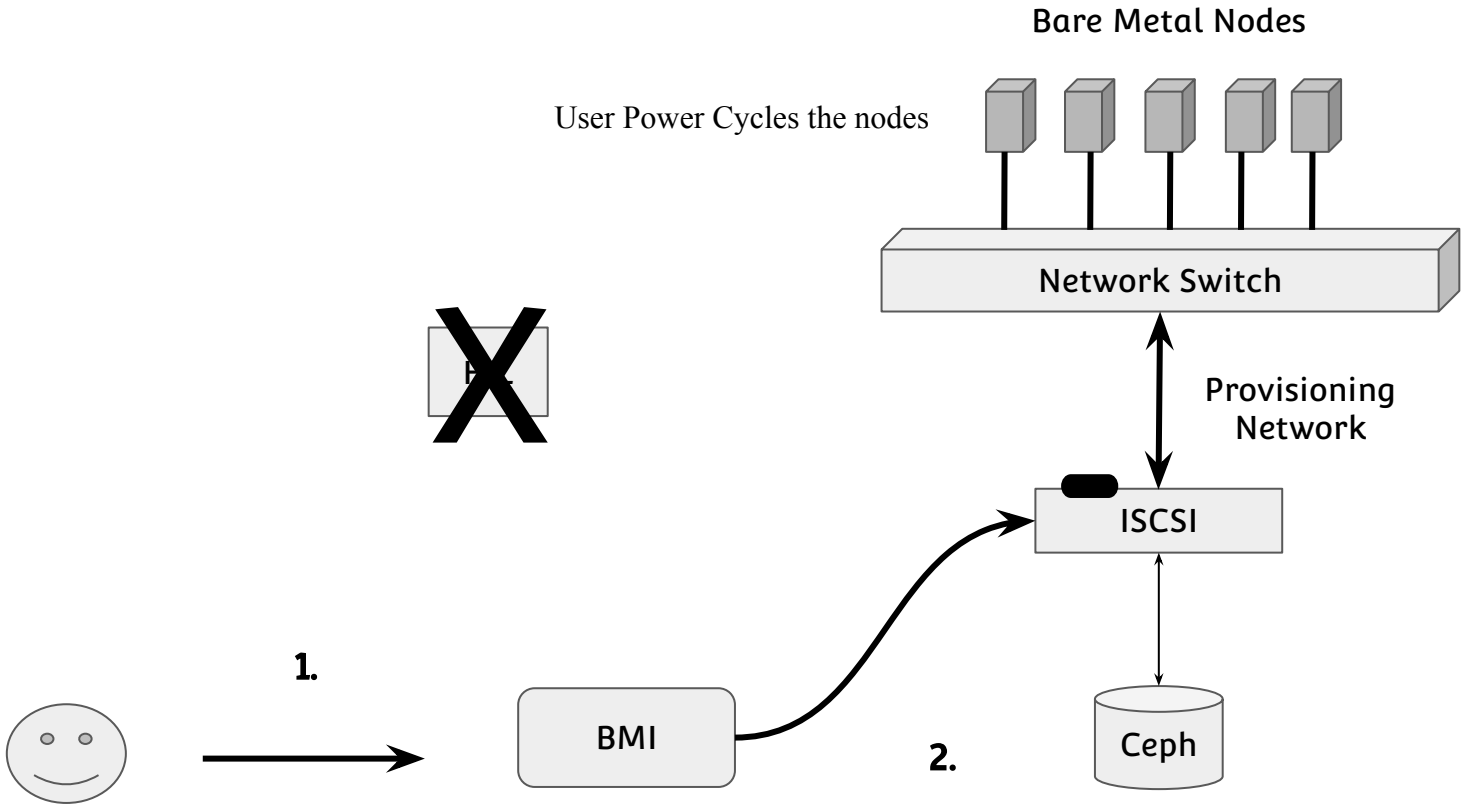
Proposed Design (Flat Provisioning Network - Single Tenant - e.g. OpenNebula)



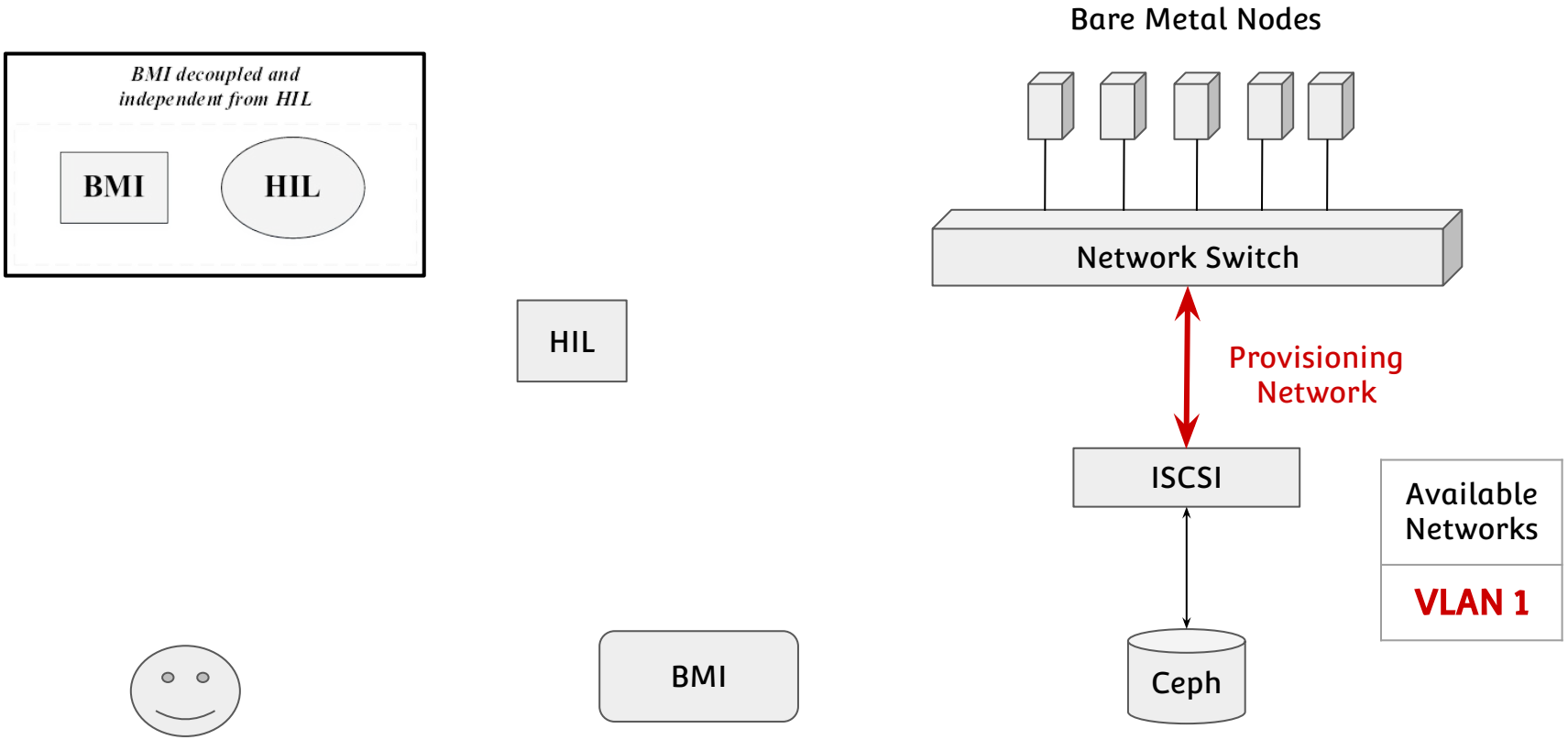
Proposed Design (Flat Provisioning Network - Single Tenant - e.g. OpenNebula)



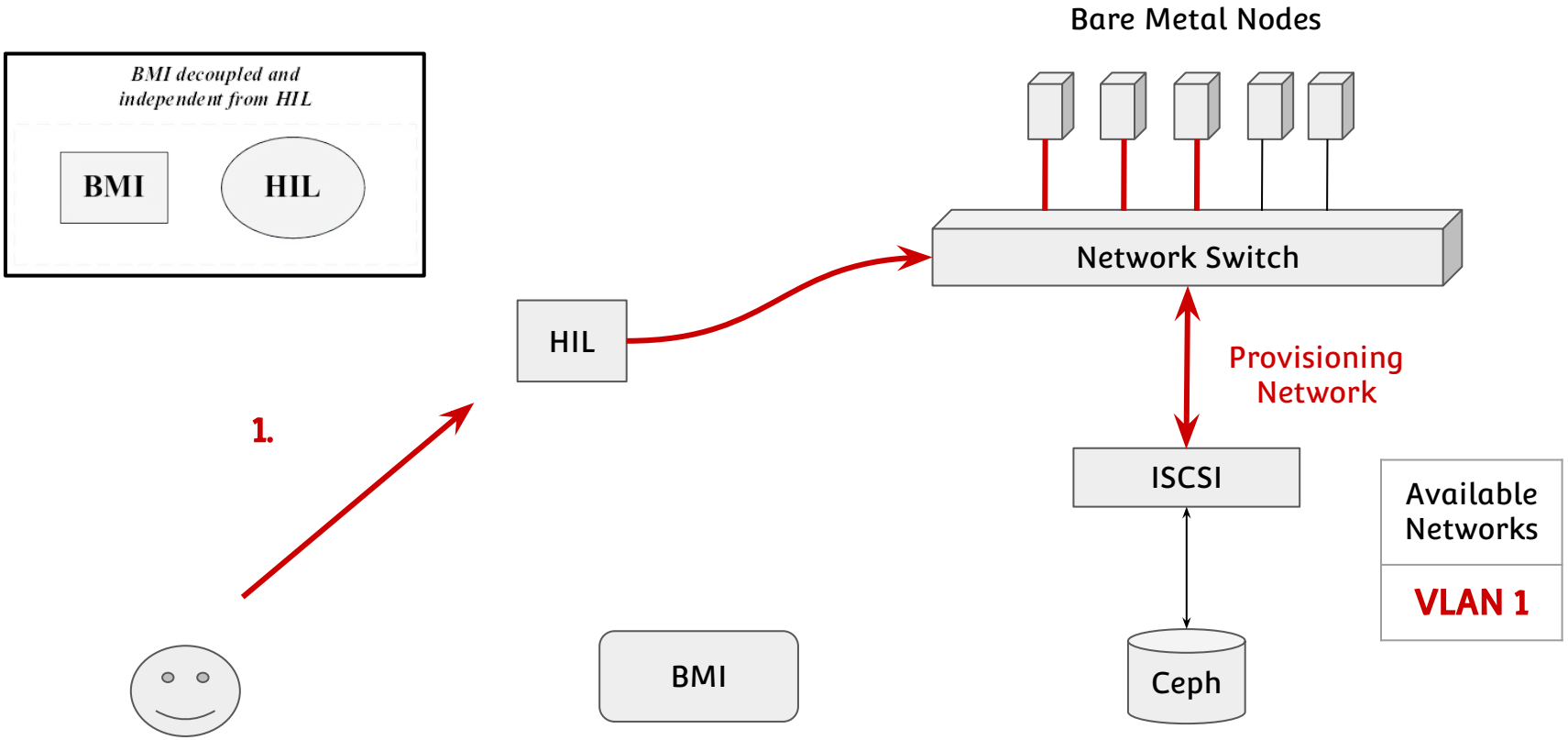
Proposed Design (Flat Provisioning Network - Single Tenant - e.g. OpenNebula)



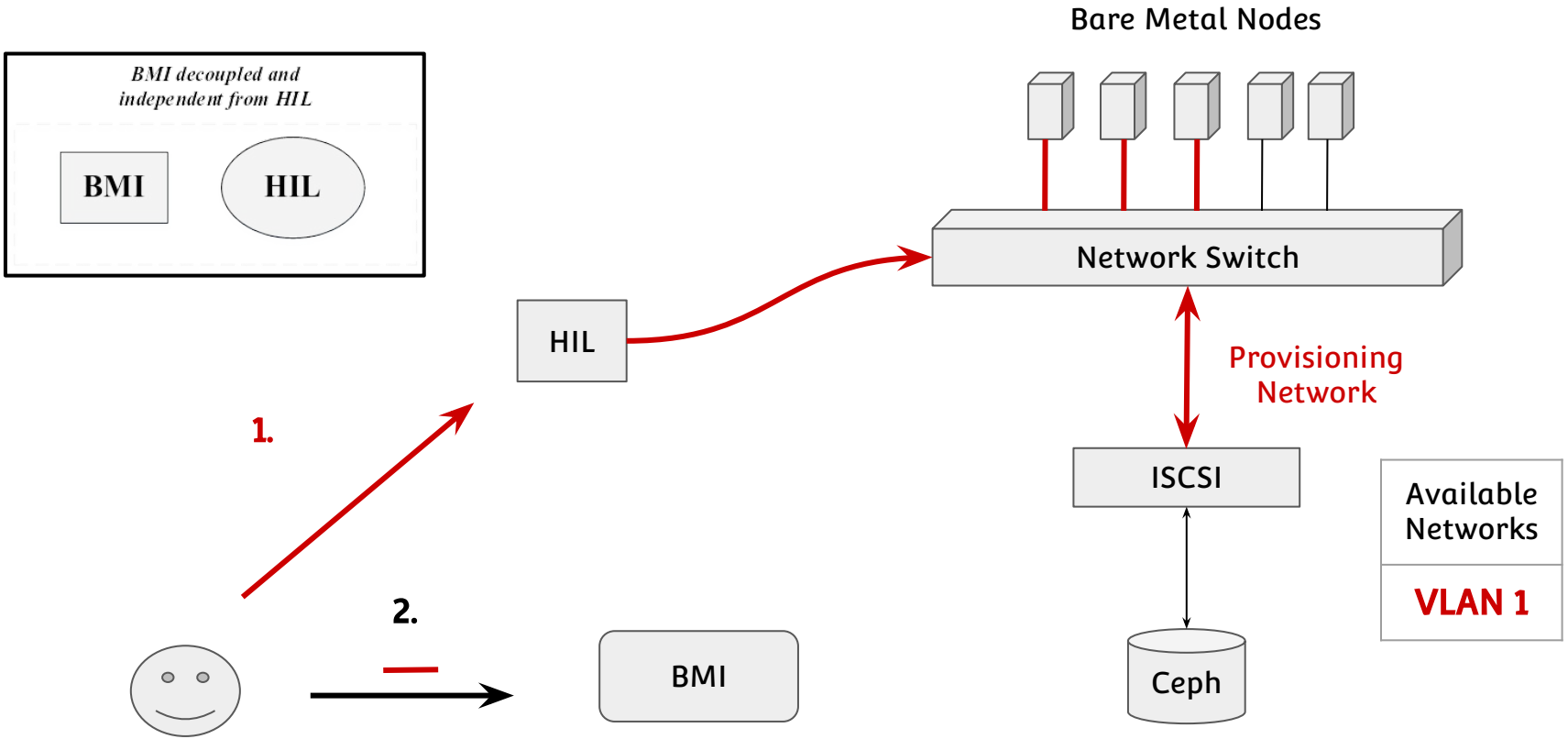
Proposed Design (Shared Provisioning Network - Trusted Tenants - e.g. MOC Research Groups)



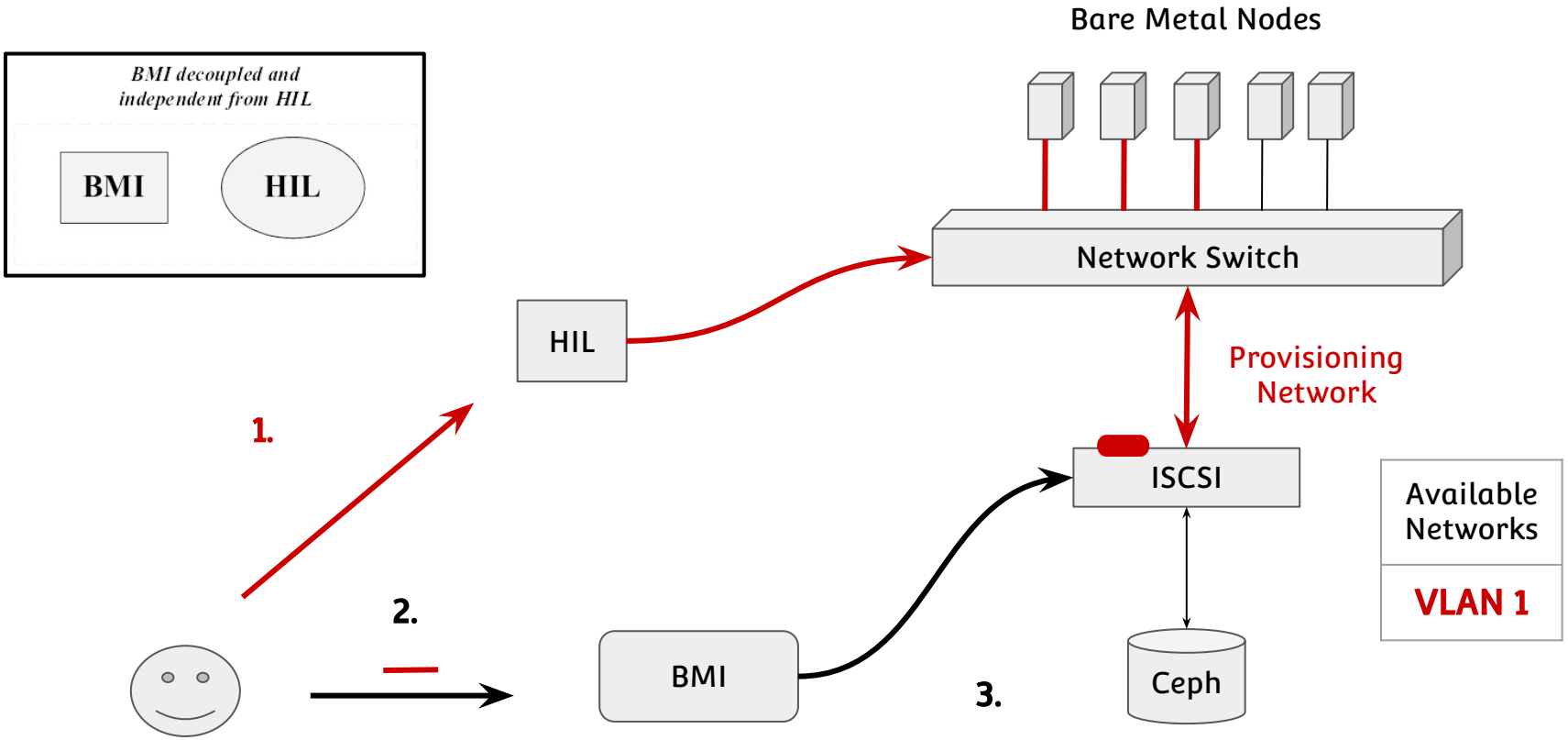
Proposed Design (Shared Provisioning Network - Trusted Tenants - e.g. MOC Research Groups)



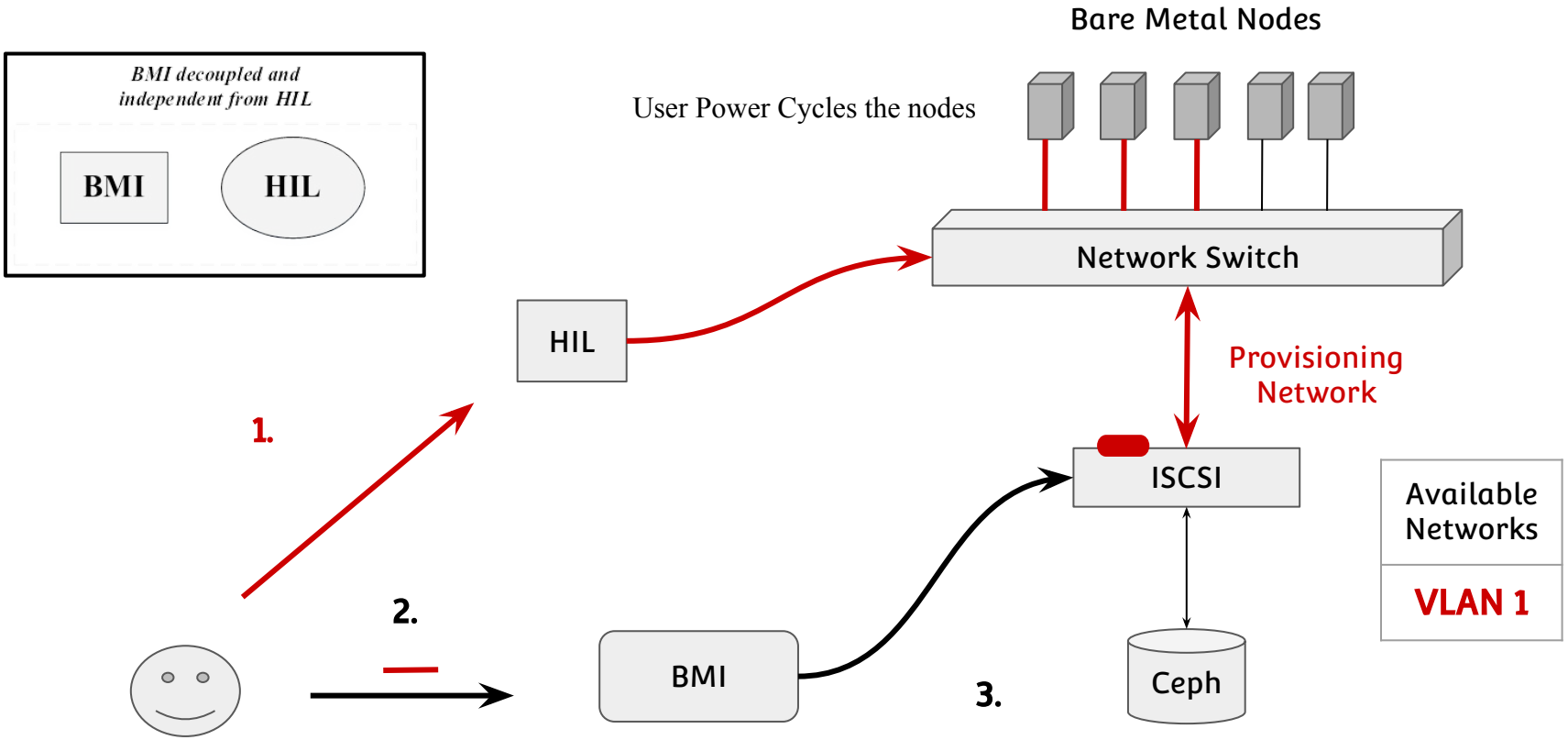
Proposed Design (Shared Provisioning Network - Trusted Tenants - e.g. MOC Research Groups)



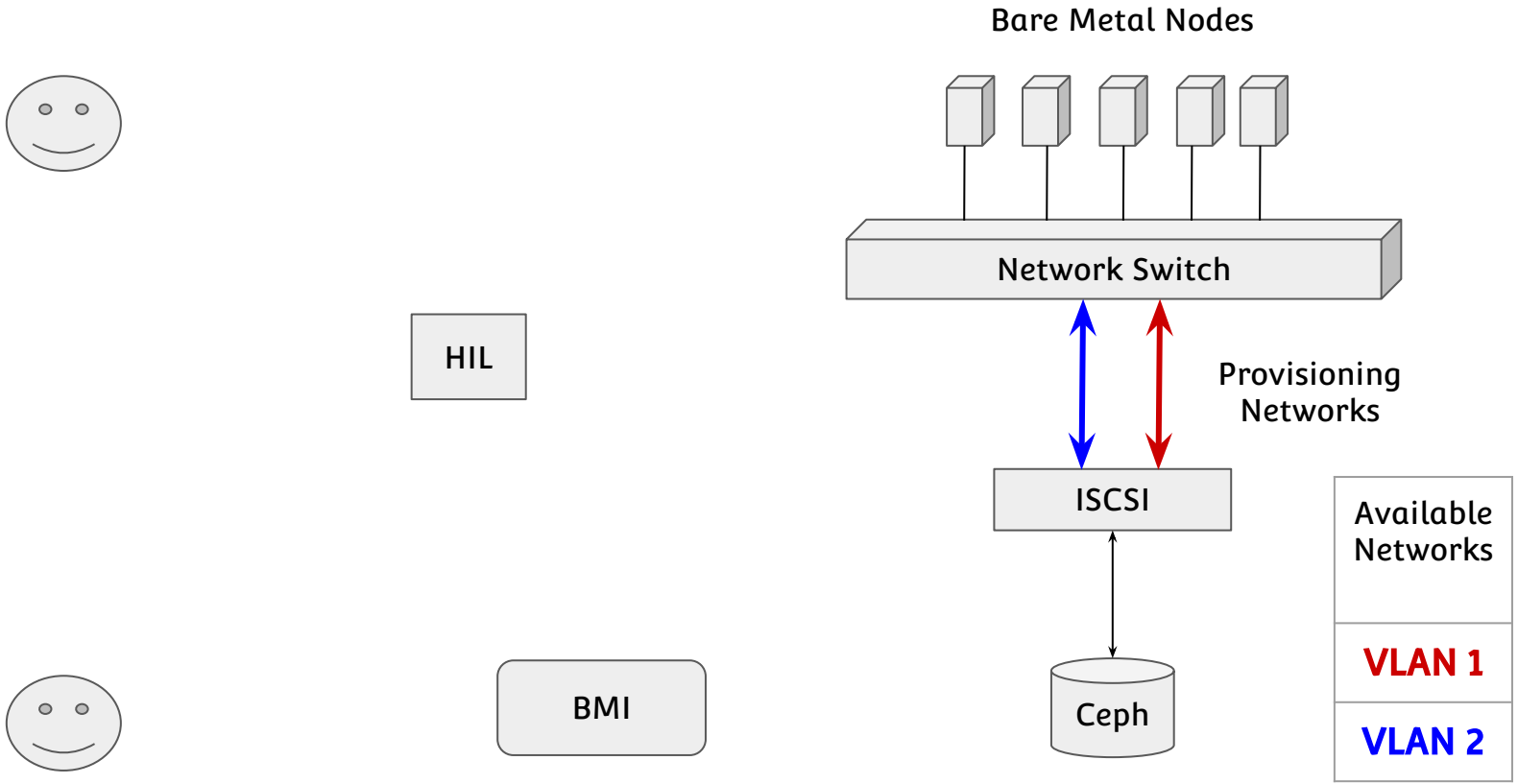
Proposed Design (Shared Provisioning Network - Trusted Tenants - e.g. MOC Research Groups)



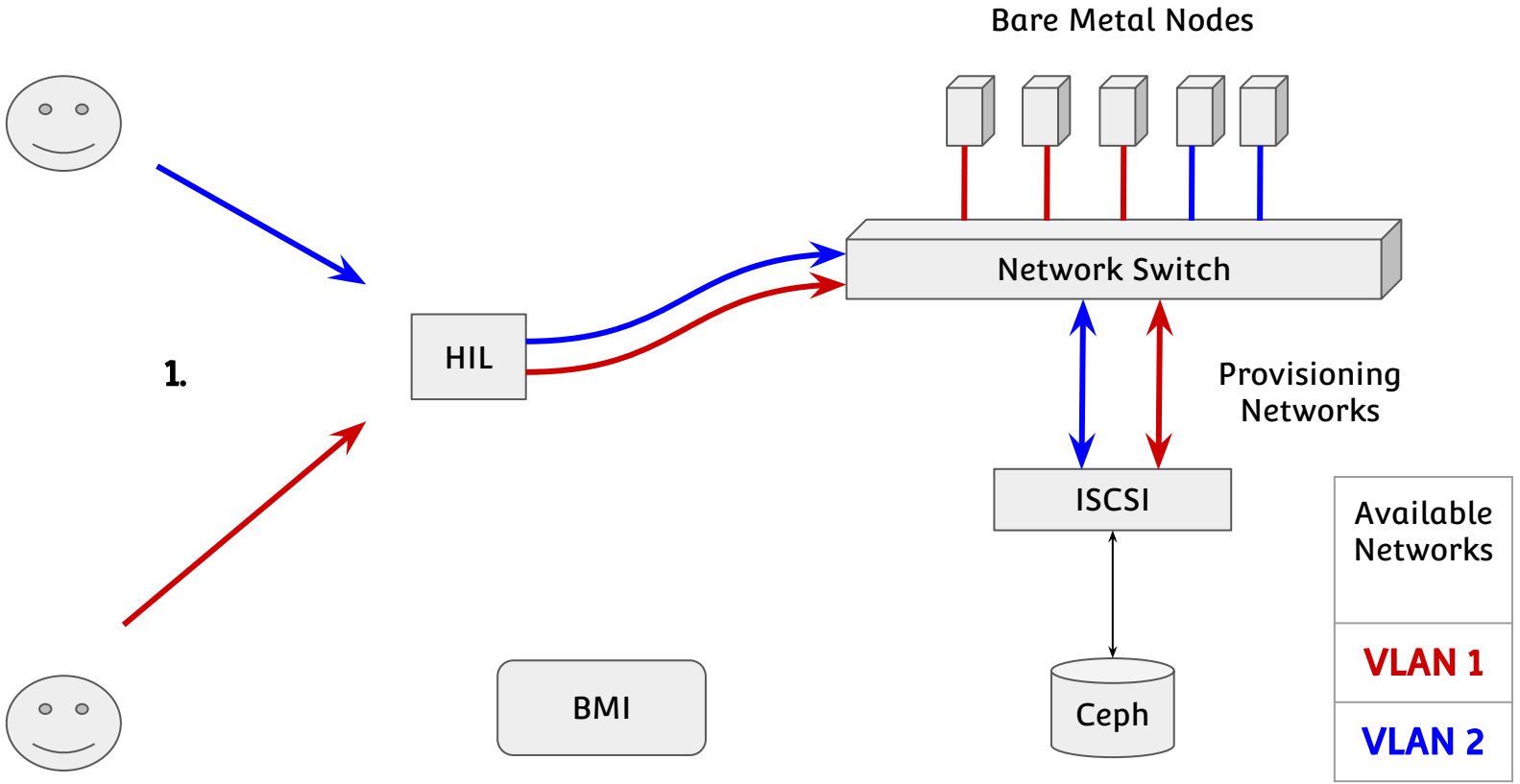
Proposed Design (Shared Provisioning Network - Trusted Tenants - e.g. MOC Research Groups)



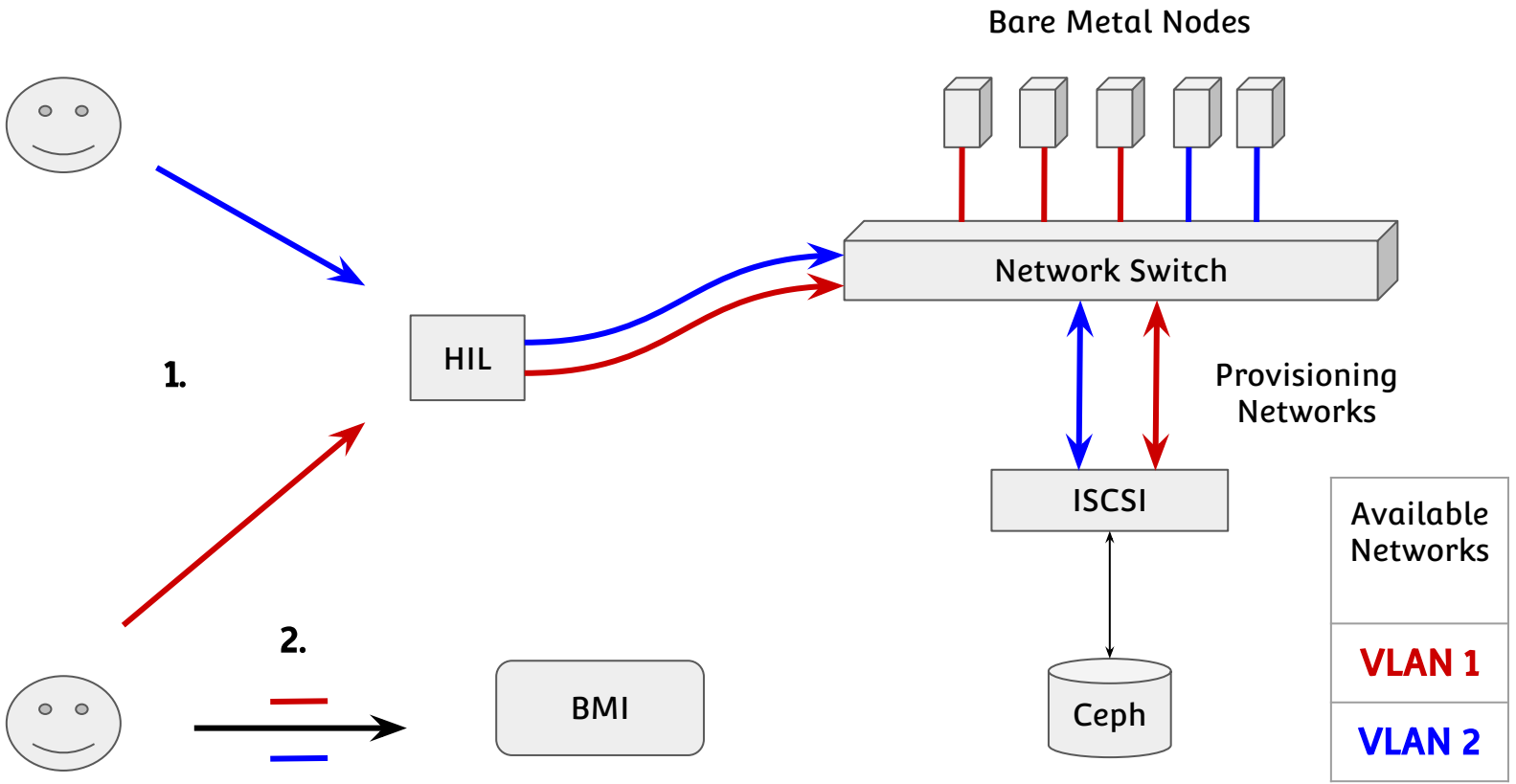
Proposed Design (Private Provisioning Networks - UnTrusted Tenants - e.g. Seccloud)



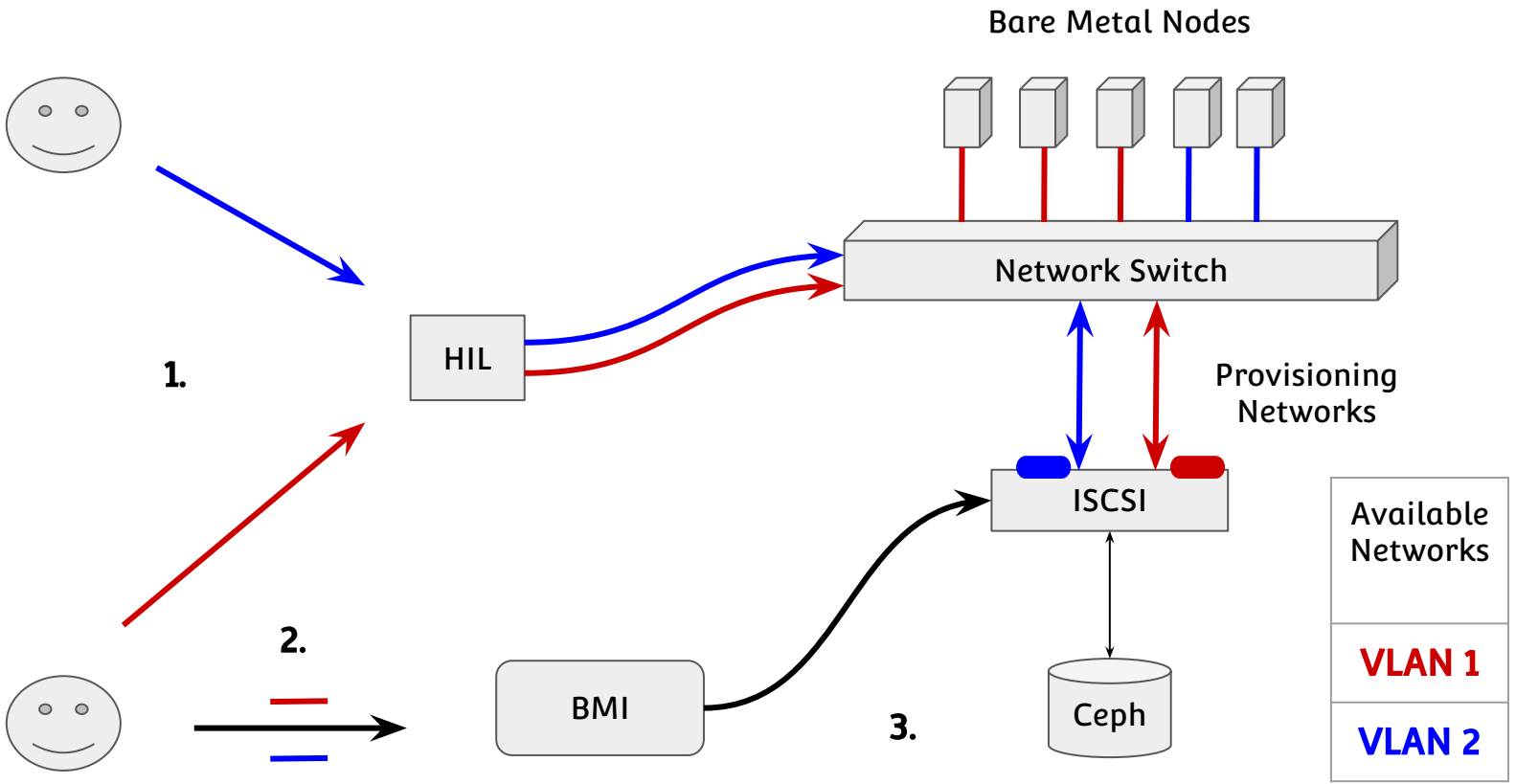
Proposed Design (Private Provisioning Networks - UnTrusted Tenants - e.g. Seccloud)



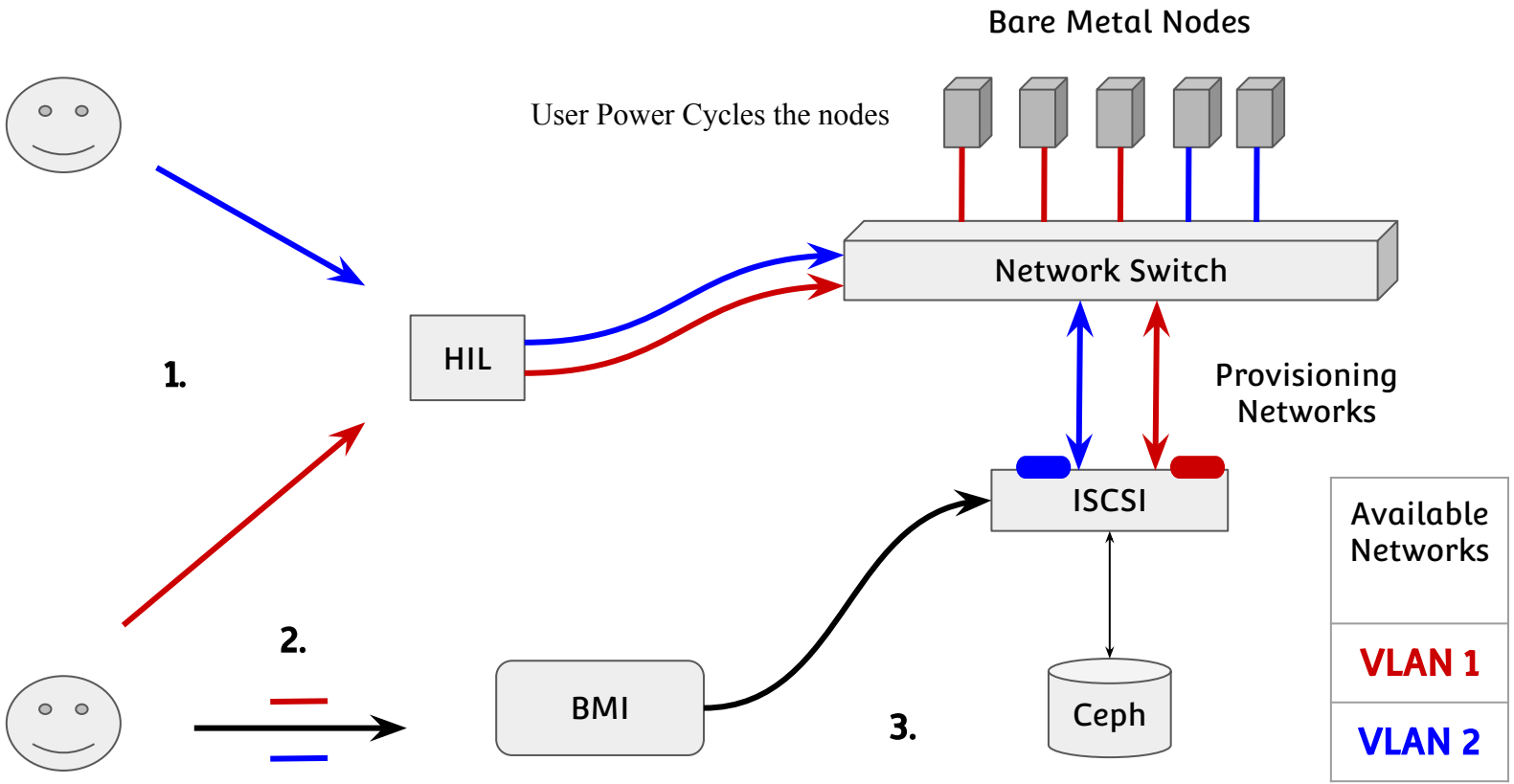
Proposed Design (Private Provisioning Networks - UnTrusted Tenants - e.g. Seccloud)

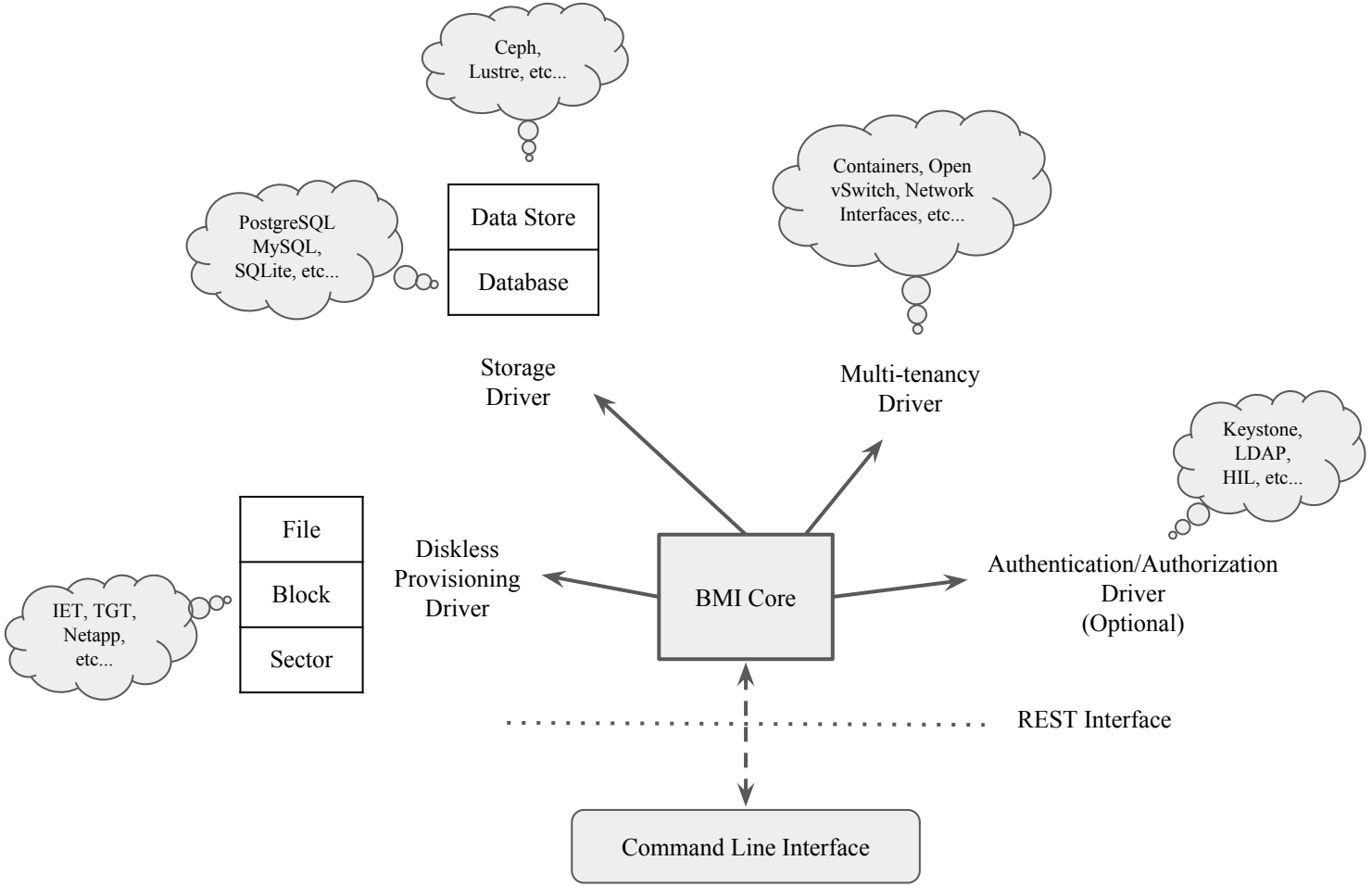


Proposed Design (Private Provisioning Networks - UnTrusted Tenants - e.g. Seccloud)



Proposed Design (Private Provisioning Networks - UnTrusted Tenants - e.g. Seccloud)




Ceph,
Lustre, etc...
PostgreSQL
MySQL,
SQLite, etc...
Data Store
Database
Containers, Open
vSwitch, Network
Interfaces, etc...
Storage
Driver
Multi-tenancy
Driver
Keystone,
LDAP,
HIL, etc...
File
Block
Sector
IET, TGT,
Netapp,
etc...
Diskless
Provisioning
Driver
BMI Core
Authentication/Authorization
Driver
(Optional)
REST Interface
Command Line Interface

The **core of BMI (**stored in the database**)**

- Defined as whatever user can manipulate
- Project, node, tag and image

The **drivers and middlewares** (have their own internal tables)

- Data store, Authorization, Diskless Provisioning, Multi-tenancy, and Authentication
  - DataStore's internal table stores the information about
    - Golden images
    - Cloned images
      - NodeID and provisioningID helps to find cloned images in the table
  - Authentication is done by a middleware
    - User connects to the middleware, and then the middleware pushes user's legitimacy to the BMI
  - How about authorization?
  - Containers for Diskless provisioning

Steps for provisioning

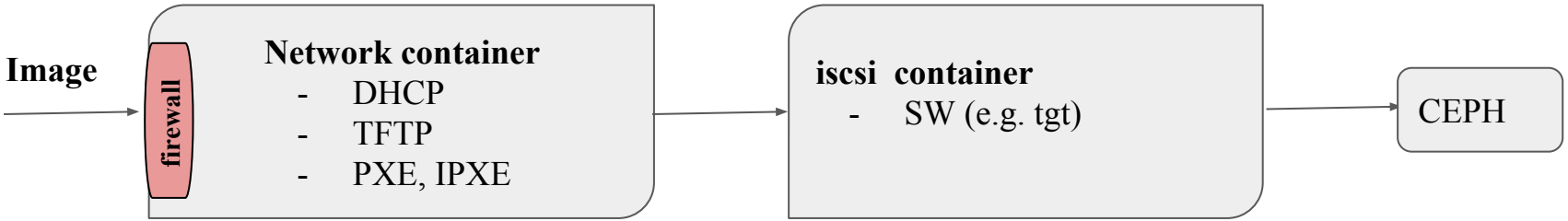
1- Prepare image → access to the DataStore's table

2- Prepare network → access to the multiTenancy's table

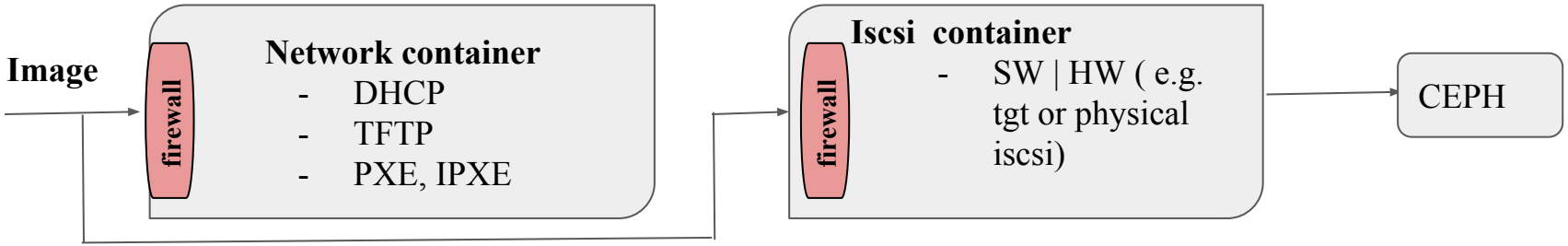
3-Expose target → access to the containers for node provisioning

Firewall is used inside the containers to whitelist specific port.

Design1)



Design 2) Better support for physical iscsi



Problem1 : Buffer Overflow in the containers → access to CEPH

Problem 2

CEPH is a pool of images, does the user see other users' images? Does CEPH expose more information than what is related to the user?