



Cesar Augusto C. Marcondes (UFSCar)
Joberto Martins (UNIFACS)
José Augusto Suruagy Monteiro (UFPE)
Kleber Vieira Cardoso (UFG)
Antônio Jorge G. Abelém (UFPA)
Vagner Nascimento (UFPA)
Iara Machado (RNP)
Tereza C.M.B. Carvalho (USP)
Charles C. Miers (USP/UDESC)
Marcos Salvador (CPqD)
Christian Esteve Rothenberg (CPqD)

Estado da Arte de
Sistemas de Controle/Monitoramento de Infraestruturas
para Experimentação de Redes de Comunicação

Mini-Curso 03 – SBRC 2012 – Ouro Preto - MG

Organização do Mini-Curso

- ▶ Motivação dos CMFs
- ▶ Ciclo de Vida de um Experimento
- ▶ Exemplos de CMFs
- ▶ Requisitos de Controle dos CMFs
- ▶ Requisitos e Funcionalidades de Monitoração dos CMFs
- ▶ Arquitetura de Federação Baseada em “Slices” (SFA)
- ▶ CMF: Protogeni – Detalhamento
- ▶ CMF: OMF – Detalhamento
- ▶ CMF: OFELIA - Detalhamento

Introdução

Cesar Marcondes (UFSCar)



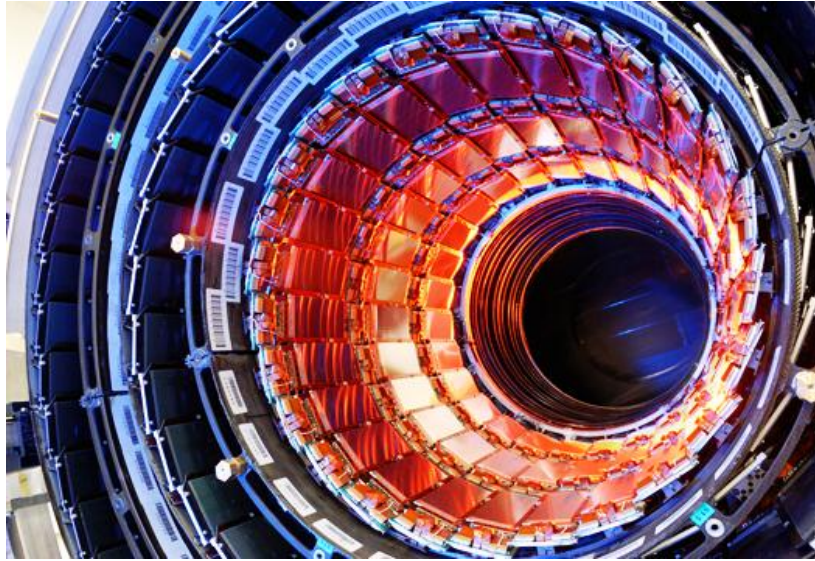
Agenda

- ▶ Motivação dos CMFs
- ▶ Ciclo de Vida de um Experimento
- ▶ Exemplos de CMFs
- ▶ Requisitos de Controle dos CMFs

Introdução

- ▶ A pesquisa em redes de comunicação, em **especial Internet do Futuro**, tem focado em inovações que seguem duas abordagens principais: clean-slate e as que têm implantação incremental
- ▶ Novas ideias de **rigorosa investigação (método científico)** em larga escala, usando uma grande quantidade de ferramentas disponíveis como simuladores, emuladores e infraestruturas para experimentação de redes
- ▶ As infraestruturas para experimentação de redes permite que as ideias sejam testadas em maior escala
- ▶ Dentro do contexto das infraestruturas, têm sido desenvolvido complexo sistemas de software, chamados de arcabouços de controle e monitoramento que habilitam experimentadores sem experiência a testar novas ideias com o mínimo de esforço.
- ▶ Objetivo: O mini-curso tem um enfoque de rever os conceitos e serviços utilizados por Arcabouços de Controle e Monitoramento, apresentar os requisitos que permeiam esse tipo de software e apresentar detalhadamente alguns CMFs desde o uso até a instalação.

Ferramentas de Experimentação para Físicos



- ▶ Pesquisador tem uma Ideia em Física Avançada
 - ▶ LHC
- ▶ Passos para Realizar a Pesquisa
 - ▶ Alocação de BEAMs
 - ▶ Escalonamento de Experimentos
 - ▶ Monitoração em Tempo Real
 - ▶ Coleta de Dados

The screenshot shows the LHC Portal website interface. At the top, there's a navigation bar with "The Portal" and "The Forum" tabs. Below this, a sidebar on the left contains various links for "Alice", "ATLAS", and other experiments, including "Web Site", "LIVE collisions", "Latest Events", "You tube", "Status", "Wiki", "Elog", "Control rm cam", "eTours", "ATLAS movie", "Episode I", "Episode II A", "Episode II B", "Not in txt Books", "Atlas built 3 M", "Web Cam", "Enews", and "Wiki". The main content area features a large circular visualization of a proton collision, showing the detector's layers and the collision point. To the right of this visualization is a smaller inset image showing a different view of the collision. At the bottom right, there's a logo for "ATLAS EXPERIMENT" and some technical details: "Run Number: 202668, Event Number: 47324388", "Date: 2012-05-02 14:00:42 CEST", and "Snapshot of a proton collision".

Grande Visão GENI (testbeds)



A bright idea

I have a great idea! The original Internet architecture was designed to connect one computer to another – but a better architecture would be fundamentally based on PEOPLE and CONTENT!

That will never work! It won't scale! What about security? It's impossible to implement or operate! Show me!

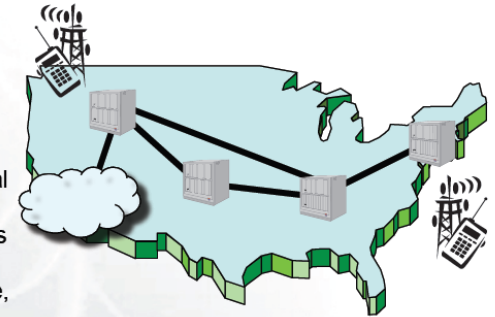


Trying it out

My new architecture worked great in the lab, so now I'm going to try a larger experiment for a few months.



And so he poured his experimental software into clouds, distributed clusters, bulk data transfer devices ('routers'), and wireless access devices throughout the GENI suite, and started taking measurements . . .

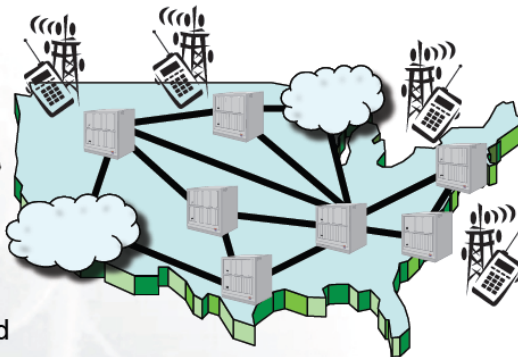


He uses a modest slice of GENI, sharing its infrastructure with many other concurrent experiments.



It turns into a really good idea

Boy did I learn a lot! I've published papers, the architecture has evolved in major ways, and I'm even attracting real users!



His slice of GENI keeps growing, but GENI is still running many other concurrent experiments.



Meanwhile . . .

I have a great idea! If the Internet were augmented with a scalable control plane and realtime measurement tools, it could be 100x as robust as it is today . . . !

And I have a great concept for incorporating live sensor feeds into our daily lives !

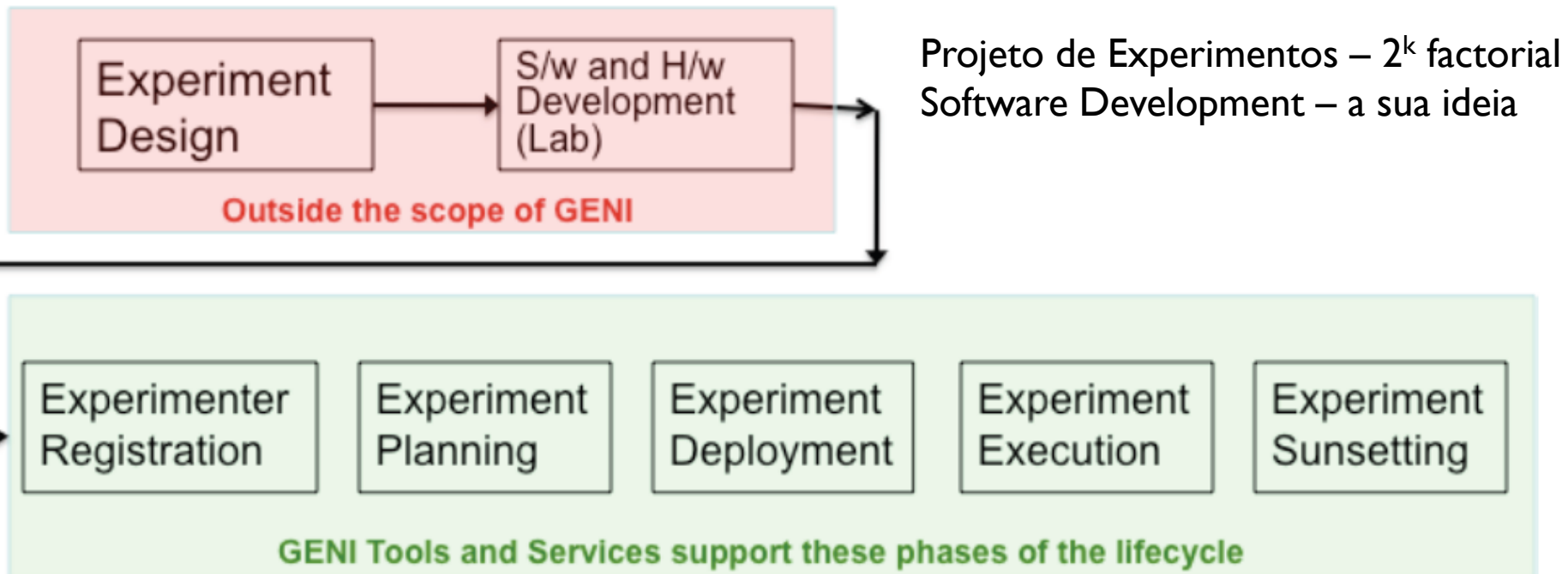


Location-based social networks are really cool!

His experiment grew larger and continued to evolve as more and more real users opted in . . .

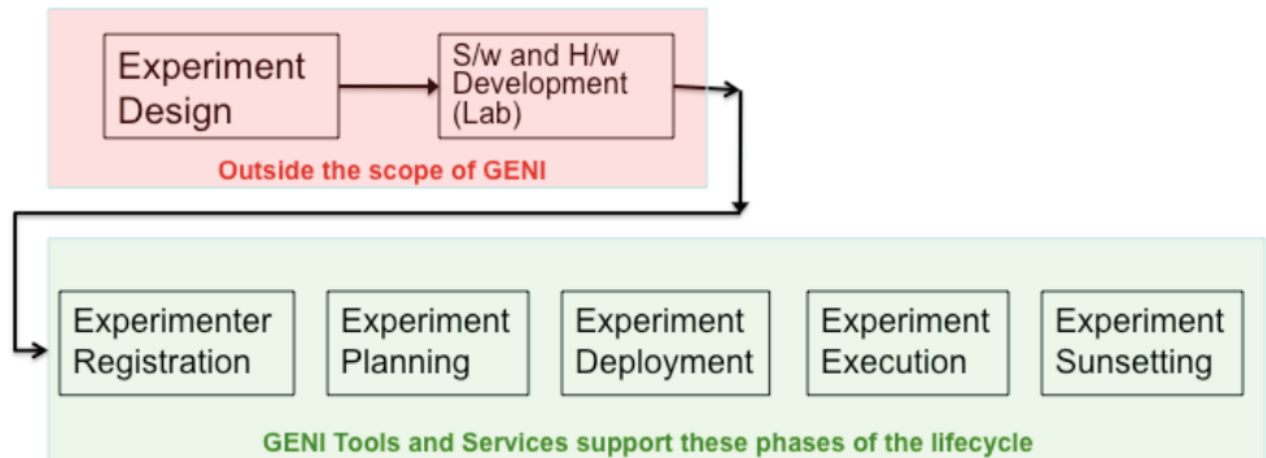
Ciclo de Vida de um Experimento

- ▶ Maioria dos CMFs – primeiro desenvolvido, GENI realizou tentativa de entender o fluxo de trabalho de um experimentador e identificar ferramentas e serviços necessários para a experimentação e extrair requisitos.

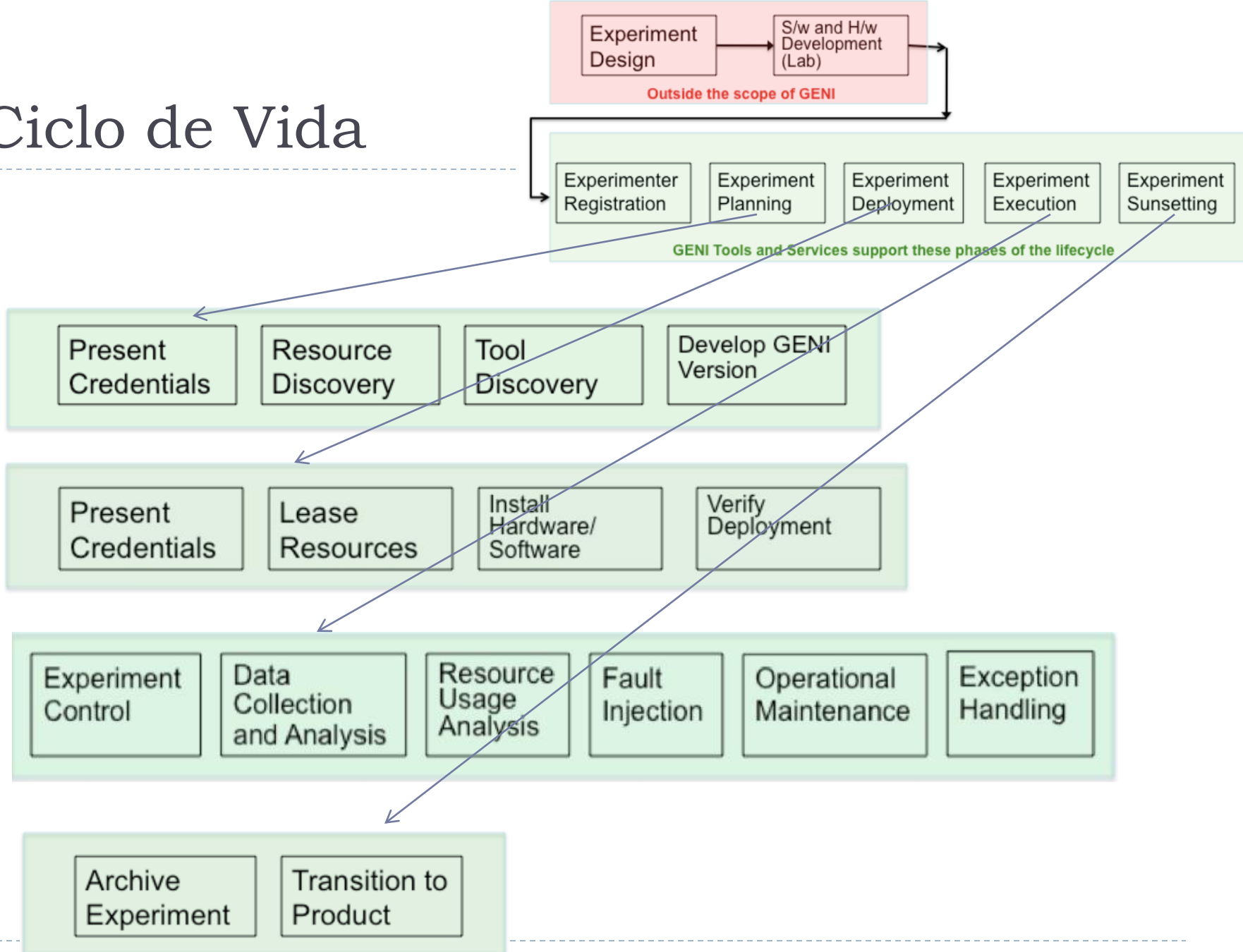


Ciclo de Vida de um Experimento

- ▶ **Experimenter Registration:** relacionado a autenticação dos usuários e dar a eles credenciais apropriadas
- ▶ **Experiment Planning:** fase onde o usuário planeja como o experimento será conduzido. Quais os recursos necessários e as ferramentas necessárias para programar estes recursos, quais os serviços de instrumentação e monitoração disponíveis.
- ▶ **Experiment Deployment:** esta relacionada com a obtenção dos recursos (por exemplo do GENI) e instalação do software e hardware para executar o experimento
- ▶ **Experiment Execution:** é a real execução do experimento: Início, Pausa, Reset, Crescimento, Redução e Parar
- ▶ **Experiment Sunsetting:** relacionado com o arquivamento do experimento de modo que esteja disponíveis para outros pesquisadores repetirem, modificarem ou estenderem o mesmo. Também inclui a ideia de transpor para ser usado por usuários reais ou tornar em produto



Ciclo de Vida



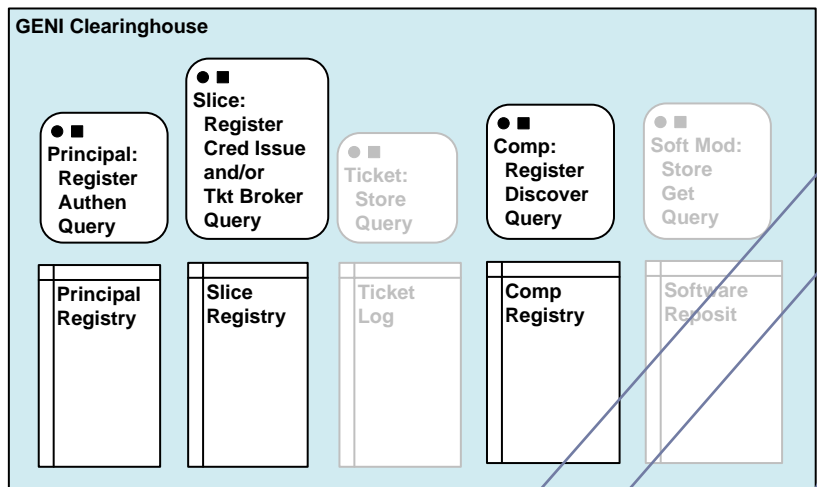
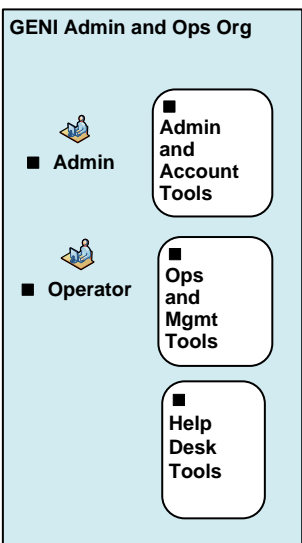
Requisitos de Controle

- ▶ Uma das questões importantes de uma testbed é o **controle dos recursos** e como **coordenar os experimentos**. Para controlar, é preciso autenticar, listar quais os recursos disponíveis e deixar disponível via API, maneiras automáticas de configuração da testbed

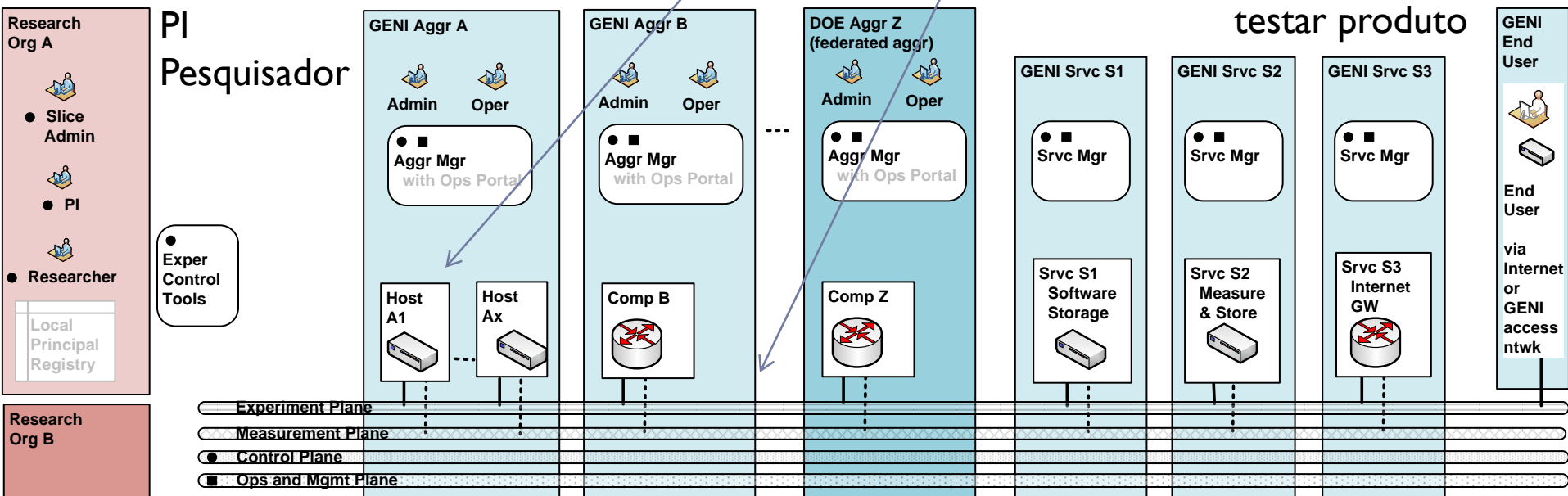
Requisitos de Controle

- ▶ Do ponto de vista de Controle é preciso ter uma infraestrutura que suporte:
 - ▶ **Programabilidade:** pesquisadores podem fazer o download de software para nós da testbed e controlar como os mesmos se comportam
 - ▶ **Virtualização e Outras Formas de Compartilhamento de Recursos:** quando possível, implementar máquinas virtuais de modo que múltiplos pesquisadores possam usar simultaneamente a infraestrutura
 - ▶ **Experimentação baseada em “Fatias de Experimentação”:** Fatias ou Slices são as frações de recursos alocados interconectados em plataformas e localizações diversas. A idéia é que os pesquisadores possam remotamente configurar, programar, debugar, operar, gerenciar e desconectar sistemas distribuídos estabelecidos pelo CMF
 - ▶ **Federação:** diferentes testbeds e seus respectivos CMFs são gerenciados e operados por diferentes organizações, e a federação é a maneira de federar todos os recursos como um grande “ecossistema” federado de experimentação

NOC



- Recursos
- Agregado de Recursos
- Planos de Controle, Monitoramento, Experimento, Gerenciamento
- ClearingHouse – centro de autenticação



Usuários podem testar produto

Exemplos de CMFs

▶ Planetlab (myPLC)

- ▶ O PlanetLab é uma plataforma de computação distribuída que foi criada e é mantida por uma comunidade de pesquisa em mais de 405 localidades em mais de 35 países. Os participantes do PlanetLab disponibilizam uma quantidade de nós (normalmente pequena, e no mínimo dois computadores) e em contrapartida podem utilizar os recursos compartilhados disponíveis em toda a plataforma para implantar e avaliar experimentos de rede em uma escala global

▶ **ProtoGENI**

▶ PrimoGENI

- ▶ O PrimoGENI é um agregado integrado ao arcabouço de controle do ProtoGENI para possibilitar a simulação de redes em tempo real. As instâncias do agregado PrimoGENI são compostas de recursos virtuais, como a rede virtual que inclui elementos simulados (roteadores, servidores, enlaces e protocolos) e elementos emulados (hosts e roteadores executando em máquinas virtuais); e os meta recursos associados à instanciação da rede virtual.

▶ ORCA/BEN

- ▶ O arcabouço de controle ORCA gerencia servidores, equipamentos de armazenamento, redes, ou outros componentes de computação em nuvem. A ideia é usar uma rede óptica experimental de escala metropolitana, para permitir ao ORCA criar e gerenciar fatias de experimentação na infraestrutura BEN em várias camadas: desde a camada física, através de DWDM (Dense Wavelength Division Multiplexing) até as camadas 2 e 3.

▶ **OMF**

▶ **OCF (OFELIA CF)**

▶ Teagle/Panlab

- ▶ Outra abordagem de Federação do PanLab, projeto análogo ao GENI

Requisitos e Funcionalidades de Monitoração dos CMFs

José Augusto Suruagy Monteiro (UFPE)

Joberto S. B. Martins (UNIFACS)



Agenda

- ▶ **Motivação**
- ▶ **Objetivos Gerais e Requisitos**
- ▶ **Usuários**
- ▶ **Arquitetura de Instrumentação e Medição (I&M) do GENI:**
 - ▶ **Serviços**
 - ▶ **Tipos de Serviços**
- ▶ **Ferramentas**

Monitoração

Aspectos Gerais

- ▶ Monitorar é uma necessidade para o experimentador
- ▶ Sistemas da Monitoração (ou monitoramento):
 - ▶ Arquitetura de Instrumentação & Medição (I&M Architecture):
 - ▶ Monitorar no contexto das redes experimentais requer um “sistema”
 - ▶ Objetivos Gerais e Requisitos de Monitoração
 - ▶ Funcionalidades, Serviços e Ferramentas
- ▶ **GENI I&M**
 - ▶ Abrangente e “Modelo de Referência” em termos desta apresentação

Monitoração - O que Monitorar?

▶ Monitoração da Infraestrutura:

- ▶ Informações de estado e operacionais da infraestrutura
- ▶ Necessário para o operador e, numa menor escala, para o experimentador:
 - ▶ Falhas
 - ▶ Status operacional dos recursos (rede, enlaces, servidores, outros) – on, off, congestionado, não disponível, outros
 - ▶ Recursos disponíveis (identificação) – lógicos e físico
 - ▶ Outros parâmetros e informações de estado

▶ Monitoração do Experimento:

- ▶ Informações decorrentes e relativas ao experimento realizado
- ▶ Análise e eventual reprodução do experimento
- ▶ Parâmetros de medição significativos para o experimentador: tempos de resposta, vazão, atrasos, parâmetros específicos ao experimento, outros

Sistemas de Monitoração

Objetivos Gerais e Requisitos

- ▶ Fornecer funcionalidades abrangentes para:
 - ▶ Coleta, Análise e Arquivamento de dados
- ▶ Facilitar a experimentação para pesquisadores (usuários) (não necessitam ser especialistas em monitoração)
- ▶ Realizar medições com precisão, acurácia, alta disponibilidade, dentre outros requisitos
- ▶ Prover informações detalhadas de desempenho
- ▶ Permitir acesso e controle de funções e serviços de medição:
 - ▶ Sensores, pontos de medição em HW e SW, MIBs, outros
- ▶ Prover segurança de forma abrangente:
 - ▶ Acesso, privacidade, disponibilização de dados, outros aspectos

Monitoração

Usuários

- ▶ Sistemas de monitoração adotam perfis de usuários de forma a tratar as questões de acesso e necessidades funcionais destes
- ▶ Usuários (modelo GENI):
 - ▶ Pesquisadores Experimentadores
 - ▶ Usuários de Experimentos
 - ▶ Operadores Centrais
 - ▶ Provedores e Operadores de Agregado
 - ▶ Provedores e Operadores de Arquivos
 - ▶ Usuários de Dados (pesquisadores)

Monitoração

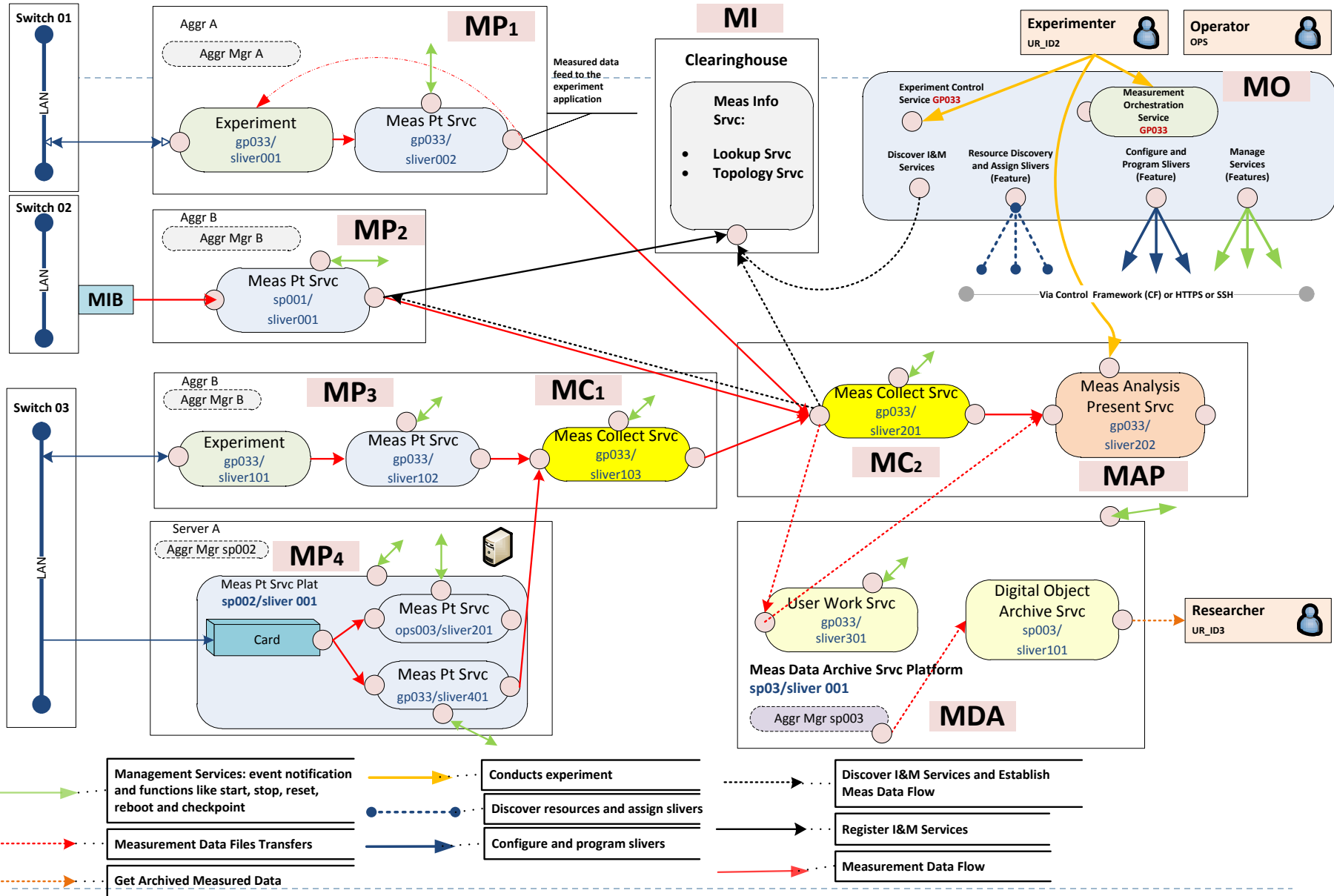
Usuários

- ▶ Pesquisadores Experimentadores
- ▶ Usuários de Experimentos
- ▶ Operadores Centrais
- ▶ Provedores e Operadores de Agregado
- ▶ Provedores e Operadores de Arquivos
- ▶ Usuários de Dados (pesquisadores)

Arquitetura de Instrumentação e Medição (I&M) do GENI

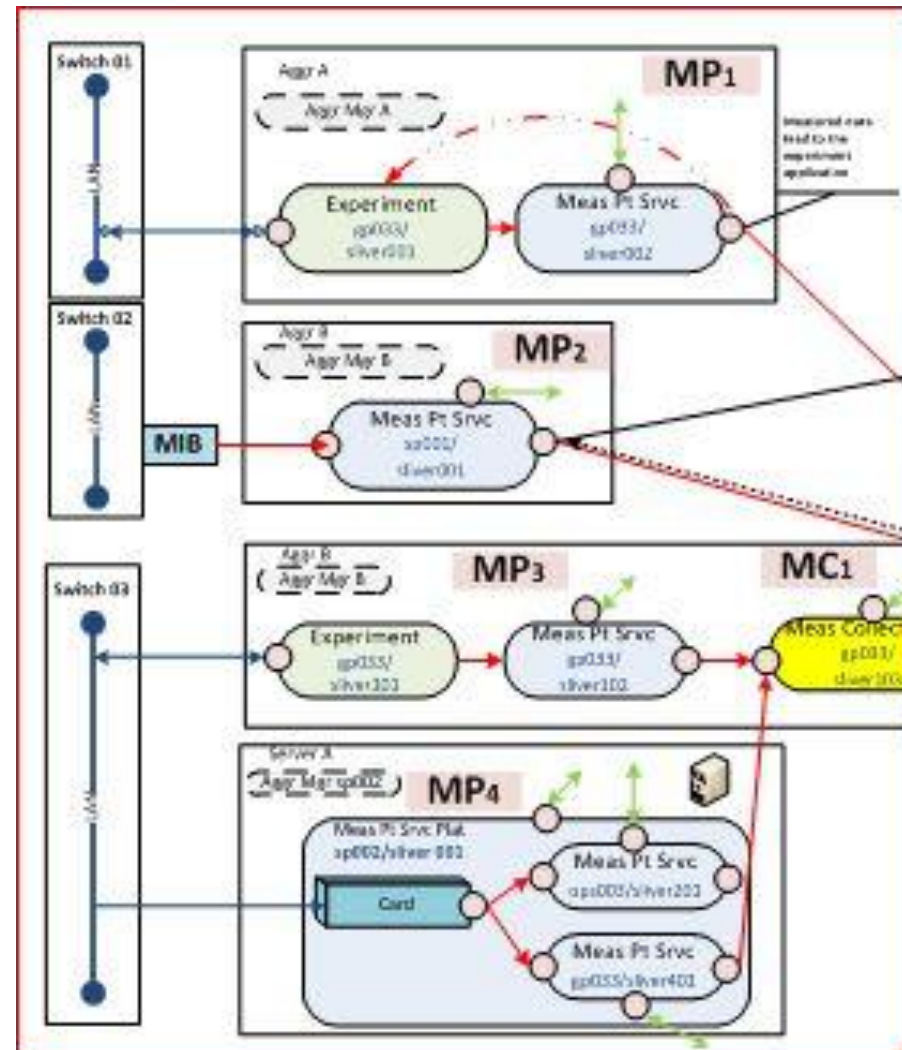
- ▶ Serviço de monitoração abrangente que pode ser adotado por diversos CMFs e outras arquiteturas de monitoração
 - ▶ Incorpora e estrutura as diversas possibilidades e elementos de monitoração
- ▶ **Serviços:**
 - ▶ Serviço de Ponto de Medição (MP)
 - ▶ Serviço de Informação de Medição (MI)
 - ▶ Serviço de Coleta de Medição (MC)
 - ▶ Serviço de Orquestração de Medições (MO)
 - ▶ Serviço de Análise e Apresentação de Medições (MAP)
 - ▶ Serviço de Arquivo de Dados de Medição (MDA)

I&M GENI - Serviços



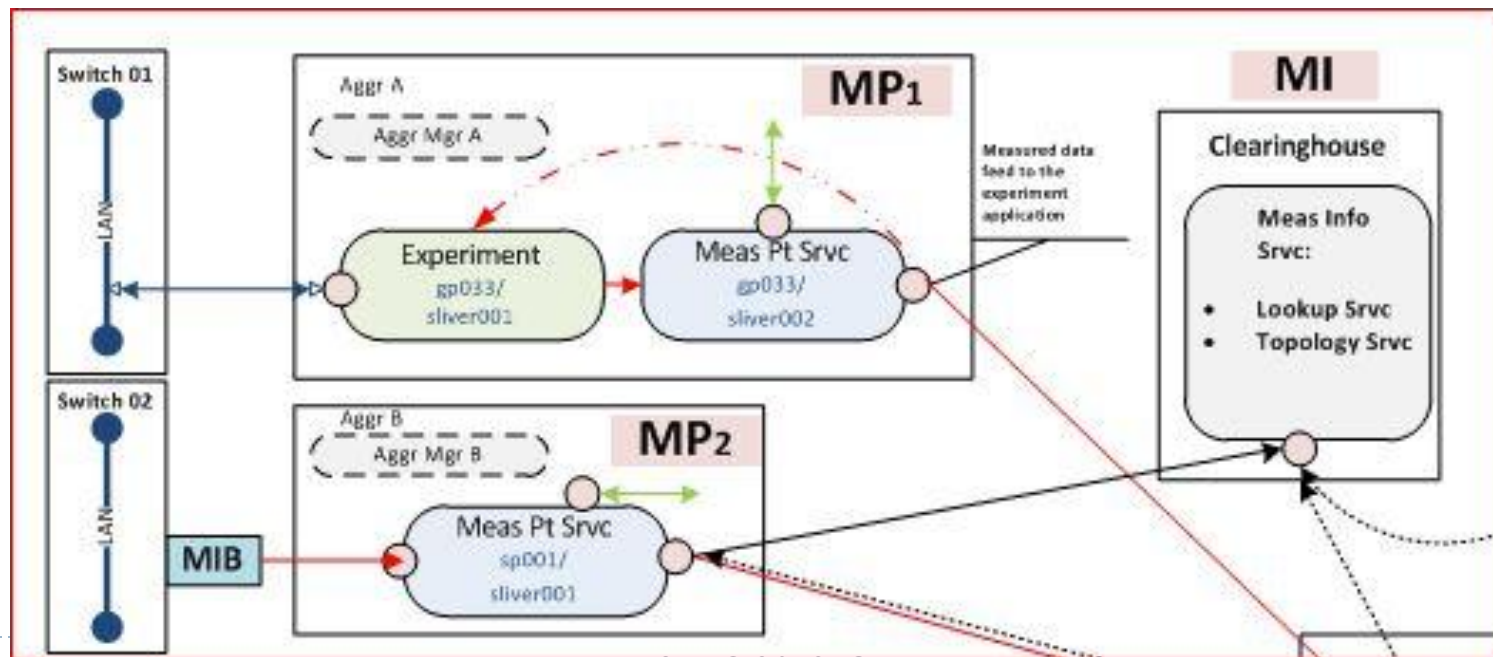
I&M GENI - Serviços

- ▶ Serviço de Ponto de Medição (MP – *Measurement Point*):
 - ▶ Provê a funcionalidade de monitoração propriamente dita (captura o dado monitorado):
 - ▶ Na infraestrutura da rede experimental, num “slice”, numa máquina virtual, ...
 - ▶ Exporta dados monitorados num determinado padrão (dependente de CMF)
 - ▶ Dados de um MP podem ser capturados e realimentados numa aplicação (comportamento adaptativo)
 - ▶ Usa diversas ferramentas e interfaces: MIBs, switches, roteadores, máquinas virtuais, placas de captura de dados, outros



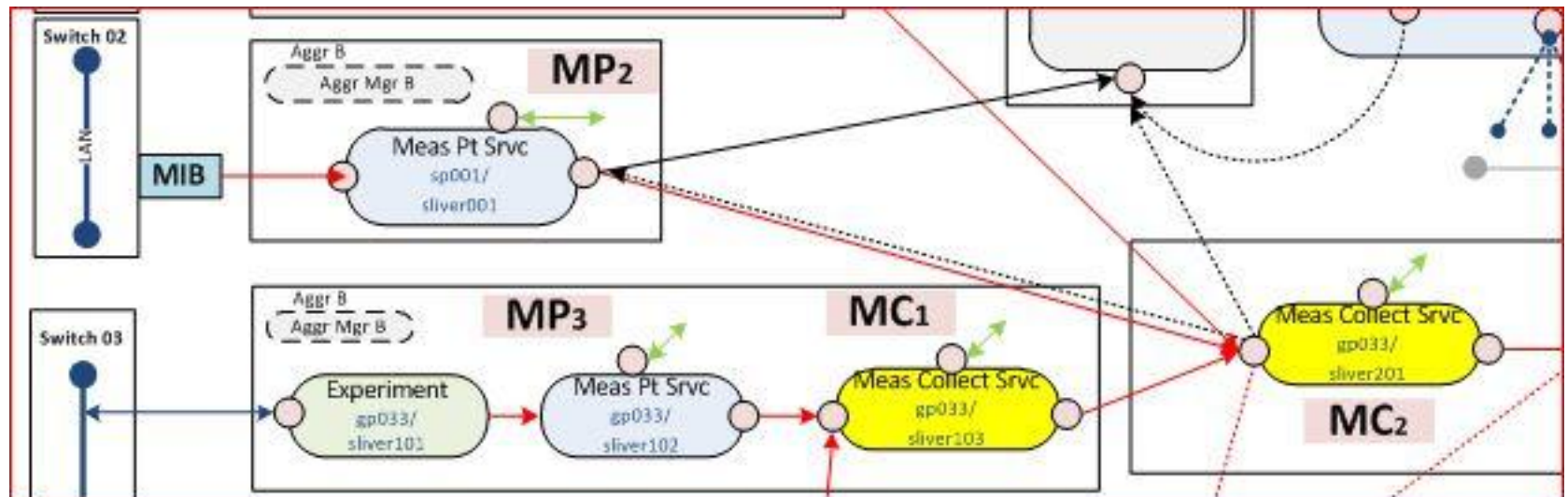
I&M GENI - Serviços

- ▶ Serviço de Informação de Medição (*MI – Measurement Information*):
 - ▶ Serviço de registro (register) e descoberta (lookup service) de dados de medição
 - ▶ Tipicamente provido por alguma forma de “Clearinghouse Service”
 - ▶ Exemplo do serviço: topologia, tipos de pontos de medição, outros



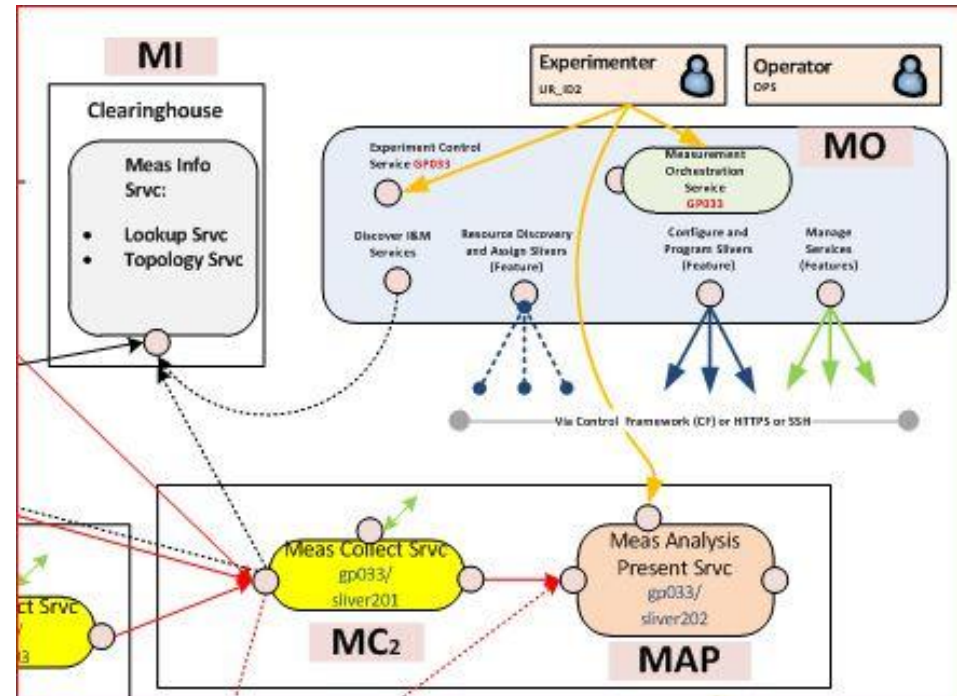
I&M GENI - Serviços

- ▶ Serviço de Coleta de Medição (MC – *Measurement Collection*):
 - ▶ Recebe como entrada dados de medições
 - ▶ Efetua algum tipo de processamento nos dados de medição:
 - ▶ Transformações, armazenamento temporário (cache), combinações de dados, outros processamentos



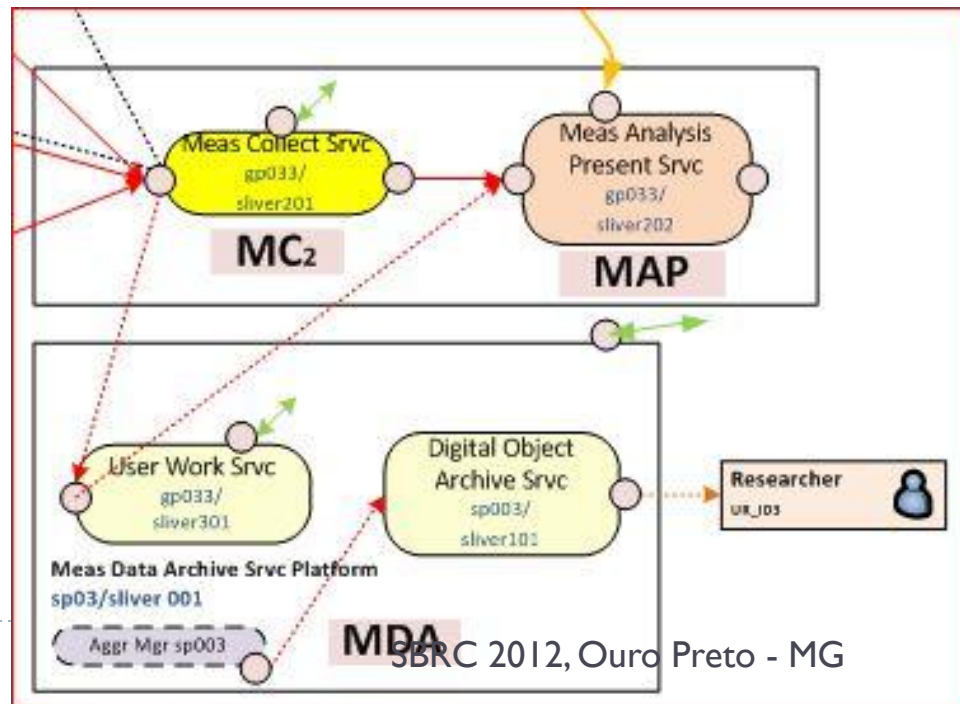
I&M GENI - Serviços

- ▶ Serviço de Orquestração de Medições (MO – *Measurement Orchestration*):
 - ▶ Faz parte do controle do experimento
 - ▶ “Orquestra” funções e serviços do sistema I&M:
 - ▶ Agrega, configura, programa e gerencia os recursos de monitoração
 - ▶ Executa a monitoração propriamente dita
 - ▶ Funcionalidade equivalente à parte de Controle e Configuração do Experimento do CMF



I&M GENI - Serviços

- ▶ Serviço de Análise e Apresentação de Medições (*MAP – Measurement Analysis and Presentation*):
 - ▶ Recebe os dados de medição
 - ▶ Analisa, processa e produz algum tipo de apresentação
- ▶ Serviço de Arquivo de Dados de Medição (*MDA – Measurement Data Archive*):
 - ▶ Serviço que provê um repositório de dados de medição para os usuários do sistema I&M



I&M GENI

Tipos de Serviços

- ▶ **4 tipos conforme a utilização**
 - ▶ Serviço em uma Fatia (“slice”)
 - ▶ Serviço comum com pedaços de componentes (“slivers”) dedicados a múltiplos experimentos
 - ▶ Serviço comum com dados compartilhados por múltiplos experimentos
 - ▶ Compartilhamento de dados através de um portal

Monitoração Ferramentas

- ▶ **GIMS – GENI I&M System:**
 - ▶ Em desenvolvimento (*WIP – Work in Progress*)
- ▶ **GEMINI (*GENI Measurement and Instrumentation Infrastructure*)**
 - ▶ Projeto GENI para estender e integrar ferramentas pré-existentes no contexto do GIMS
 - ▶ LAMP – *Leveraging and Abstraction Measurements with PerfSONAR*
 - ▶ INSTOOLS - *Instrumentation Tools for a GENI Prototype*

Monitoração – Ferramentas

LAMP

- ▶ Sistema de instrumentação e medição baseado no perfSONAR
- ▶ Ferramentas suportadas pelo LAMP:
 - ▶ OWAMP, BWCTL, Ganglia, PingER, NTP; ps-BUOY MA; outras
- ▶ LAMP adapta perfSONAR-PS ao modelo de autenticação e autorização do GENI
 - ▶ Acrescentou anotação e configuração distribuída como o UNIS (*Unified Information Service*)

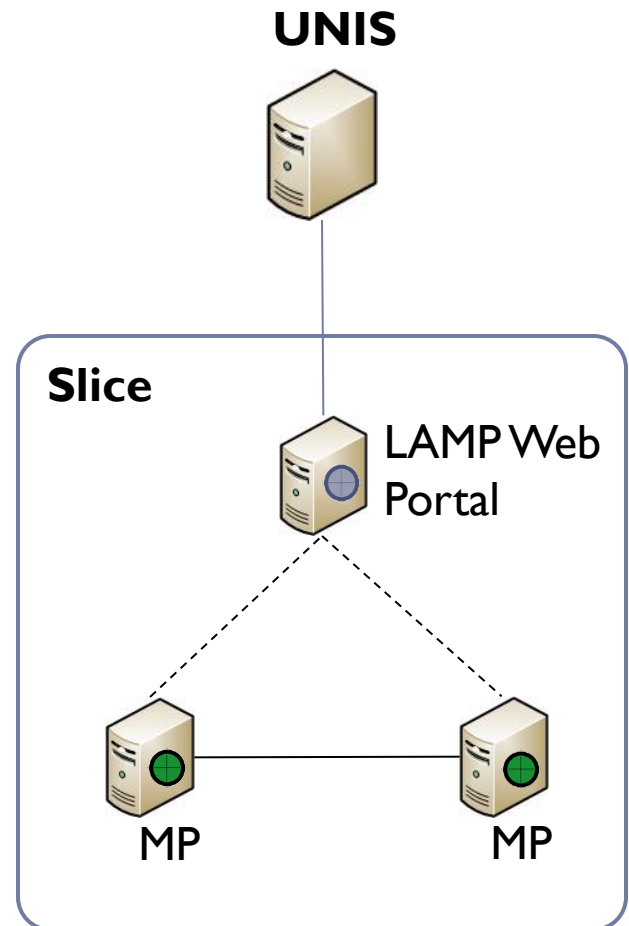
Monitoração – Ferramentas

PerfSONAR

- ▶ *Middleware* que interage com ferramentas de monitoração e dados de medição (diversos formatos)
- ▶ Disponibiliza os mesmo através de interface padrão orientada a serviços:
 - ▶ Utiliza padrão NM-WG (*Network Monitoring Working Group*) do OGF (*Open Grid Forum*)

Monitoração – Ferramentas - LAMP

- ▶ **Lamp Portal:**
 - ▶ Utilizado pelos experimentadores para gerenciar e visualizar os serviços e dados I&M
- ▶ **UNIS:**
 - ▶ Combinação de um serviço de descoberta (lookup) e serviço de topologia
- ▶ **MPs:**
 - ▶ Nós com ferramentas perfSONAR instaladas (OWAMP, BWCT, outras)



Monitoração – Ferramentas

INSTOOLS

- ▶ Instrumentação para o ambiente do usuário (Emulab e ProtoGENI)
- ▶ Composto por pontos de medição sobretudo passivas (MPs) instalados em fatias (slices)
- ▶ Captura, armazena e processa dados de medição (diversas formas) (MC)
- ▶ Disponibiliza através de portal

Software	Information
SNMPd	Routing Table
	IP Traffic
	ICMP Traffic
	TCP Traffic
	UDP Traffic
	CPU Utilization
	Memory Utilization
	Total Network Traffic
	Link-specific Traffic
	Link-specific Unicast Traffic
ssh/arp	ARP cache
ssh/netstat	TCP streams
ssh/netstat	UDP listeners
ssh/ps	Process list
ssh/lsmmod	Installed Kernel Modules

GEMINI Próximas Etapas

- ▶ Combinação da interface e portal do INSTOOLS com os padrões e definições de metadados do perfSONAR/LAMP.
- ▶ Novos desenvolvimentos:
 - ▶ Registro Global como extensão do UNIS
 - ▶ Funcionalidade de Publicação e Assinatura
 - ▶ Ampliação das medições
 - ▶ Integração com o OpenFlow
 - ▶ Novas ferramentas de visualização e armazenamento

Monitoração – Ferramentas

GIMI

- ▶ *large scale GENI Instrumentation and Measurement Infrastructure*
- ▶ Objetivo: prover ferramentas fáceis de usar baseadas na OML (*Orbit Measurement Library*)

Monitoração – Ferramentas

GIMI: OML

- ▶ Coleta de dados em tempo real num ambiente de larga escala
- ▶ Componentes:
 - ▶ Serviço OML
 - ▶ Biblioteca de cliente
 - ▶ Servidor de coleta de dados
- ▶ Está prevista a exportação dos dados através de uma interface perfSONAR:
 - ▶ compartilhamento de dados com o GEMINI

Monitoração – Ferramentas TopHat

- ▶ **Componente de medições ativas do PlanetLab Europa através do MySLICE.**
 - ▶ Serviço de monitoração de topologia durante todo o ciclo de vida de um experimento
 - ▶ A partir destes dados os usuários podem selecionar recursos com características desejadas de conectividade e atrasos.
- ▶ **Utiliza ferramentas tais como:**
 - ▶ DIMES, ETOMIC, SONoMA, Team Cymru.

Arquitetura de Federação Baseada em “Slices” (SFA)

Kleber Cardoso (UFG)



Agenda

- ▶ **Conceito de federação**

- ▶ **SFA 2.0**
 - ▶ Atores
 - ▶ Entidades
 - ▶ Abstrações
 - ▶ Tipos de dados
 - ▶ Interfaces e operações

Federação

- ▶ Alocação unificada de recursos em diferentes testbeds para executar experimentos
- ▶ **Motivação**
 - ▶ Aumentar escala e heterogeneidade, economizar através compartilhamento, melhorar a qualidade da pesquisa (testes em ambientes mais próximos do real)
- ▶ **Dificuldades**
 - ▶ Tratar diferentes tipos de recursos (heterogeneidade de hardware/software) e testbeds (heterogeneidade de regras/políticas)

Atores

- ▶ **Proprietários**
 - ▶ Responsáveis por estabelecer políticas de uso de alto nível para seus equipamentos
- ▶ **Operadores**
 - ▶ Responsáveis por manter a plataforma executando
- ▶ **Pesquisadores (e desenvolvedores)**
 - ▶ Responsáveis por executar os experimentos
- ▶ **Âncoras de identidade (ou provedores de identidade, antigos investigadores principais – PI)**
 - ▶ Responsáveis por autorizar pesquisadores (desenvolvedores) individuais a acessar a infraestrutura

Mediação

▶ Proprietários

- ▶ Declaração de políticas para alocação e uso de recursos para infraestruturas sobre seu controle e fornecimento de mecanismos para fazer cumprir essas políticas

▶ Operadores

- ▶ Gerenciamento do substrato de rede, o que inclui instalação de novo hardware e remoção de antigo ou com defeito, instalação e atualização de software e monitoração da rede em termos de desempenho, funcionalidade e segurança

▶ Pesquisadores

- ▶ Criação e preenchimento de fatias de experimentação, reserva de recursos para elas e execução experimentos nelas

▶ Âncoras de identidade

- ▶ Identificação do conjunto de pesquisadores de uma organização que possuem permissão para utilizar uma infraestrutura

Entidades

- ▶ **Autoridade de gerenciamento (management authority – MA)**
 - ▶ Responsáveis por algum subconjunto de componentes do substrato
- ▶ **Autoridade de fatia de experimentação (slice authority – SA)**
 - ▶ Responsáveis pelo comportamento de um conjunto de fatias de experimentação
- ▶ **Usuário**
 - ▶ Ator que representa um ou mais papéis dentro de uma infraestrutura (proprietário, operador, pesquisador ou âncora de identidade)

Abstrações (1)

▶ Componente

- ▶ Bloco de construção primário da arquitetura
- ▶ Coleção de recursos que podem ser multiplexados entre múltiplos usuários
 - ▶ físicos (exemplo: CPU, memória, disco)
 - ▶ Lógico (exemplo: descritor de arquivo, número de portas)
 - ▶ Sintético (exemplo: caminho para encaminhamento rápido de pacote)

▶ Gerenciador de agregado (aggregate manager – AM)

- ▶ Exporta uma interface bem definida e acessível remotamente para os componentes disponíveis em uma testbed

▶ Gerenciador de componente (component manager – CM)

- ▶ Um caso especial de um AM que responde por apenas um componente

Abstrações (2)

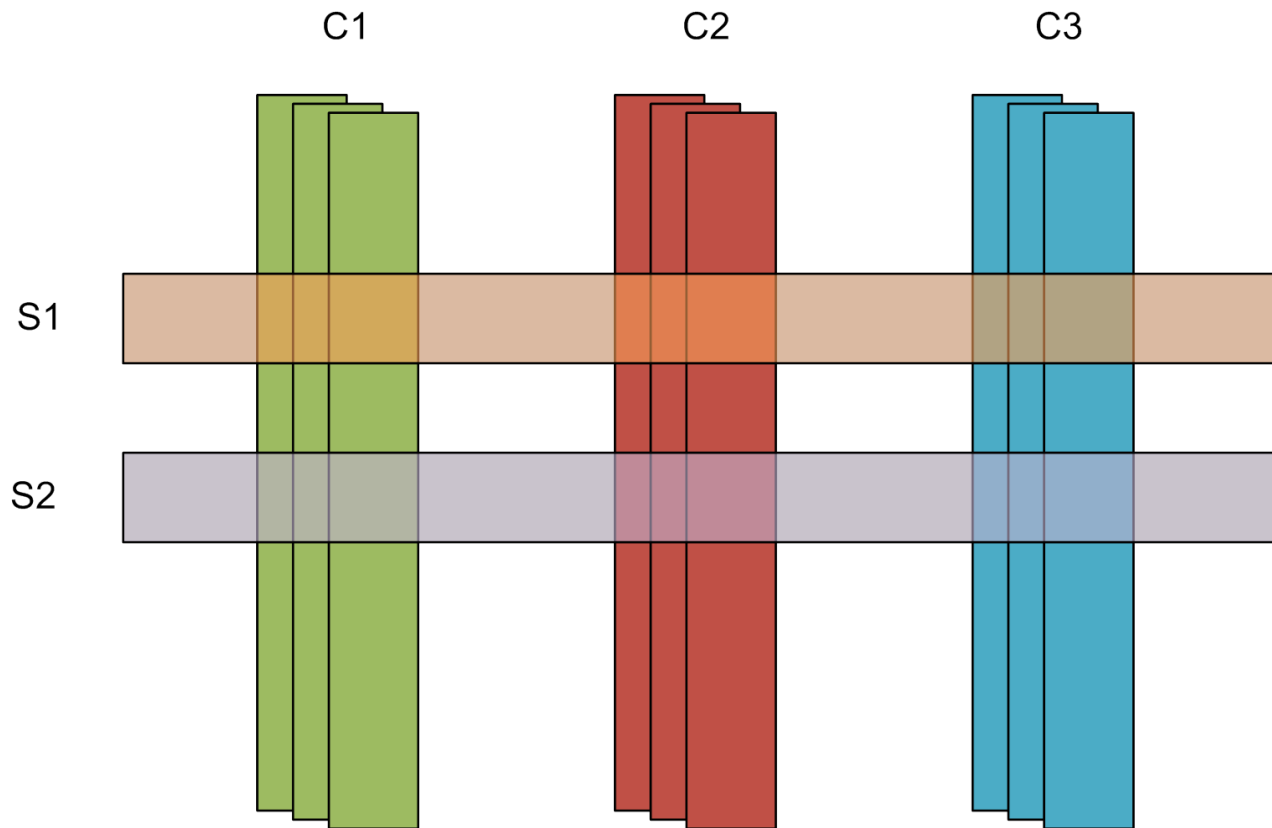
▶ Fatia de experimentação (slice)

- ▶ Perspectiva de um pesquisador: uma rede de recursos de computação e comunicação capaz de executar um experimento ou serviço de rede experimental federada
- ▶ Perspectiva de um operador: abstração primária para fins de contabilidade

▶ Porção (sliver)

- ▶ Parte de um componente que é isolado através de particionamento e/ou virtualização de recursos desse componente
- ▶ O componente deve fornecer mecanismos para isolar as slivers umas das outras

Componentes, fatias e porções



Identificadores e nomes

- ▶ **Certificado: GID = (PublicKey, UUID, Lifetime)**
 - ▶ Há um para cada objeto: componente, fatia, MA, SA, usuário
 - ▶ Universally Unique Identifier permanece o mesmo, ainda que a chave pública seja alterada
 - ▶ Base para um sistema seguro, no qual uma entidade é capaz de confirmar sua legitimidade e verificar a autenticidade de um objeto

- ▶ **Registro: HRN (Human-Readable Name)**
 - ▶ Mapeia nomes para certificados
 - ▶ Identifica a sequência de autoridades responsáveis por um objeto
 - ▶ top-level_authority.sub_authority.sub_authority.name

Tipos de dados

- ▶ **RSpec (Resource Specification)**
 - ▶ Não especificada pela SFA

- ▶ **Registry Record**
 - ▶ Record = (HRN, GID, Type, Info)

- ▶ **Ticket**
 - ▶ Ticket = (RSpec, GID, SeqNum)

- ▶ **Credential**
 - ▶ Credential = (CallerGID, ObjectGID, ObjectHRN, Expires, Privileges, Delegate)

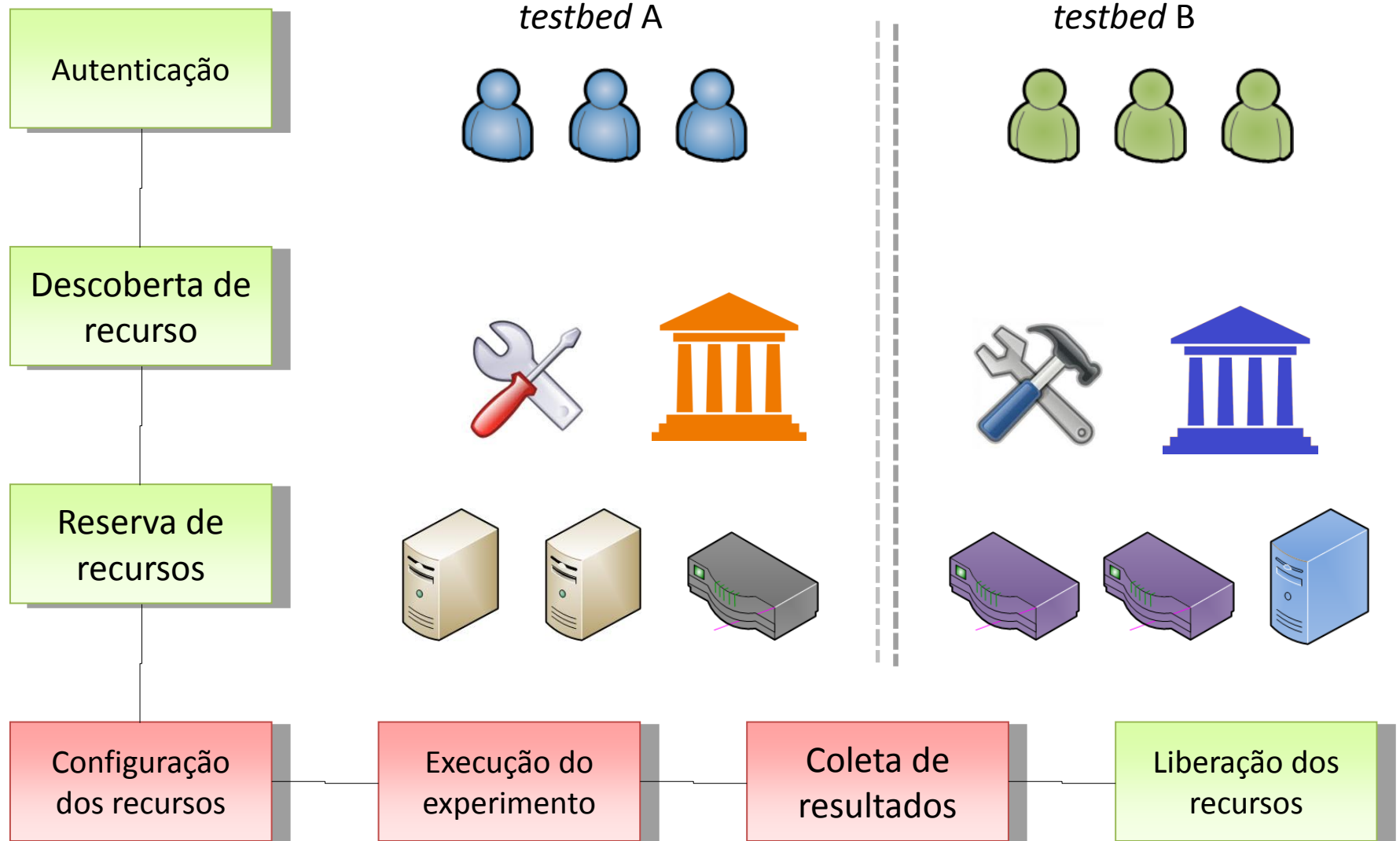
interfaces and operations

Interfaces	Operations
Slice	GetTicket, RedeemTicket, ReleaseTicket, CreateSlice, SplitTicket, LoanResources, UpdateSlice, StopSlice, StartSlice, ResetSlice, DeleteSlice, ListSlices, GetResources, GetSliceBySignature
Management	SetBootState, GetBootState, Reboot
Registry	Register, Remove, Update, Resolve, List, GetCredential

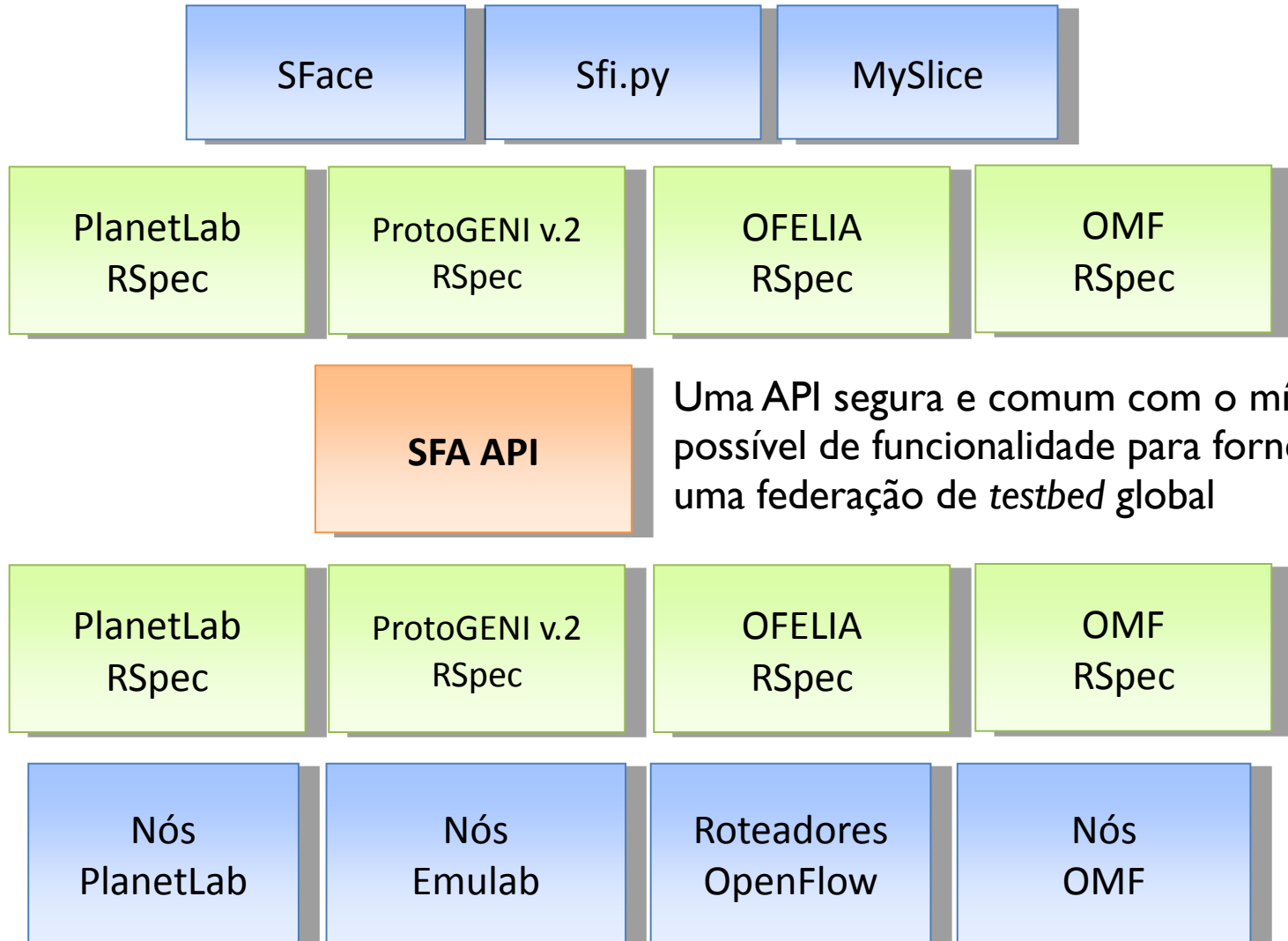
Resumo sobre SFA

- ▶ Nomes (fatias, usuários, componentes, entidades)
 - ▶ Estrutura hierárquica
- ▶ Autenticação e autorização
 - ▶ Certificados X.509 e credenciais assinadas
 - ▶ Troca de certificados
- ▶ Descrição dos recursos (RSpecs)
 - ▶ Apenas a linguagem (XML), não a semântica
- ▶ API padrão (XMLRPC)
 - ▶ Para gerenciar objetos (fatias, usuários, componentes, entidades)
 - ▶ Para gerenciar recursos (divulgar, procurar, solicitar)
 - ▶ Para gerenciar fatias (criar, remover, iniciar, apagar)

SFA e o ciclo de experimentação



Ampulheta SFA



Uma API segura e comum com o mínimo possível de funcionalidade para fornecer uma federação de *testbed* global



CMF: Protogeni Detalhamento

Marco A. Torrez Rojas (USP)

Tereza C. M. B. Carvalho, Charles C. Miers (USP)



Agenda

- ▶ **Parte I - Emulab**
 - ▶ Emulab LARC/USP
 - ▶ Como funciona o Emulab
- ▶ **Parte II – ProtoGENI**
 - ▶ O que é o ProtoGENI
 - ▶ Arcabouço de Controle e Monitoramento ProtoGENI
 - ▶ *Backbone* ProtoGENI
 - ▶ Federação ProtoGENI
- ▶ **Parte III – Demonstração**

Agenda

- ▶ **Parte I - Emulab**
 - ▶ **Emulab LARC/USP**
 - ▶ **Como funciona o Emulab**
- ▶ **Parte II – ProtoGENI**
 - ▶ O que é o ProtoGENI
 - ▶ Arcabouço de Controle e Monitoramento ProtoGENI
 - ▶ *Backbone* ProtoGENI
 - ▶ Federação ProtoGENI
- ▶ **Parte III – Demonstração**

Emulab

▶ *Testbed* Experimental

- ▶ Desenvolvido para a realização de pesquisa experimental em redes e sistemas distribuídos, provê emuladores de rede, simuladores e testes com redes internet

▶ Principios do Projeto

- ▶ Transparencia
- ▶ Virtualização
- ▶ Automação
- ▶ Eficiencia

▶ Recursos Providos

- ▶ Nós locais (PCs)
 - ▶ nós de borda, nós sem fio, radio definido por software (GNU Radio)
- ▶ Enlace de rede virtuais
 - ▶ topologias arbitrárias, enlaces de rede local e de rede WAN

Emulab

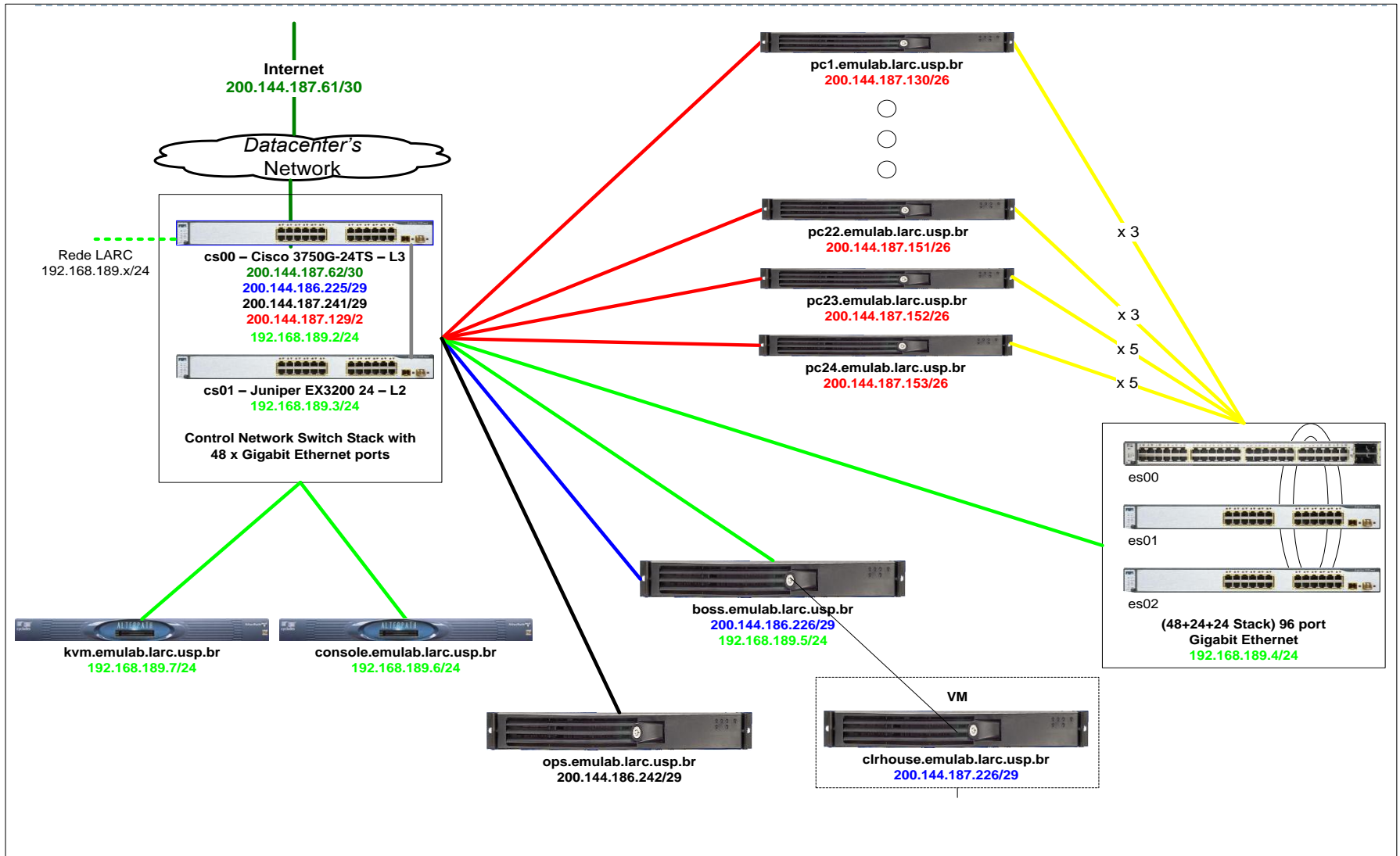
- ▶ Principal *Testbed* Emulab
- ▶ Universidade de Utah
 - ▶ Cerca de 500 nos PC;
 - ▶ Projetado para suportar pesquisa e ensino



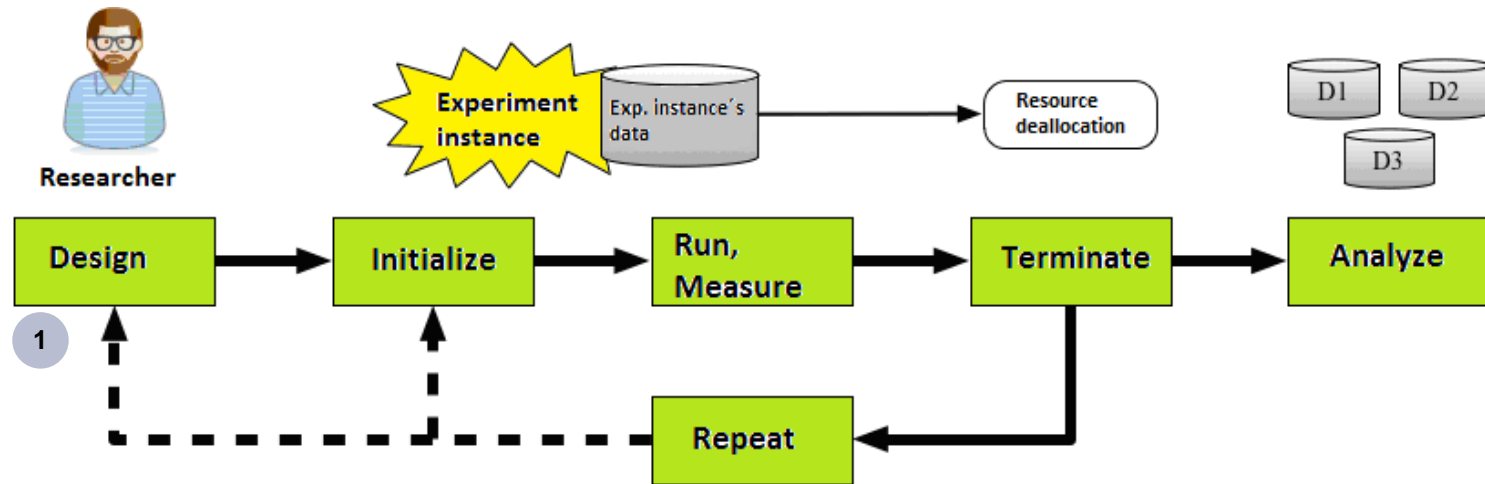
Emulab LARC/USP

- ▶ **Financiado por: Fapesp / Kyatera & Ericsson**
 - ▶ 24 nós para experimentos
 - ▶ 17 Intel Pentium 4 (3GHz), 3GB RAM (DDR2), 80GB HD
 - ▶ 7 Xeon Quad-Core 2.4Ghz, 4GB RAM, 500 GB HD
 - ▶ Todos com interfaces 3 x Gb Ethernet para experimentos
 - ▶ 3 servidores
 - ▶ Armazenamento para dados dos usuários / imagens de SO / Clearinghouse
 - ▶ 5 switches Gigabit (Cisco e Juniper)

Emulab LARC/USP

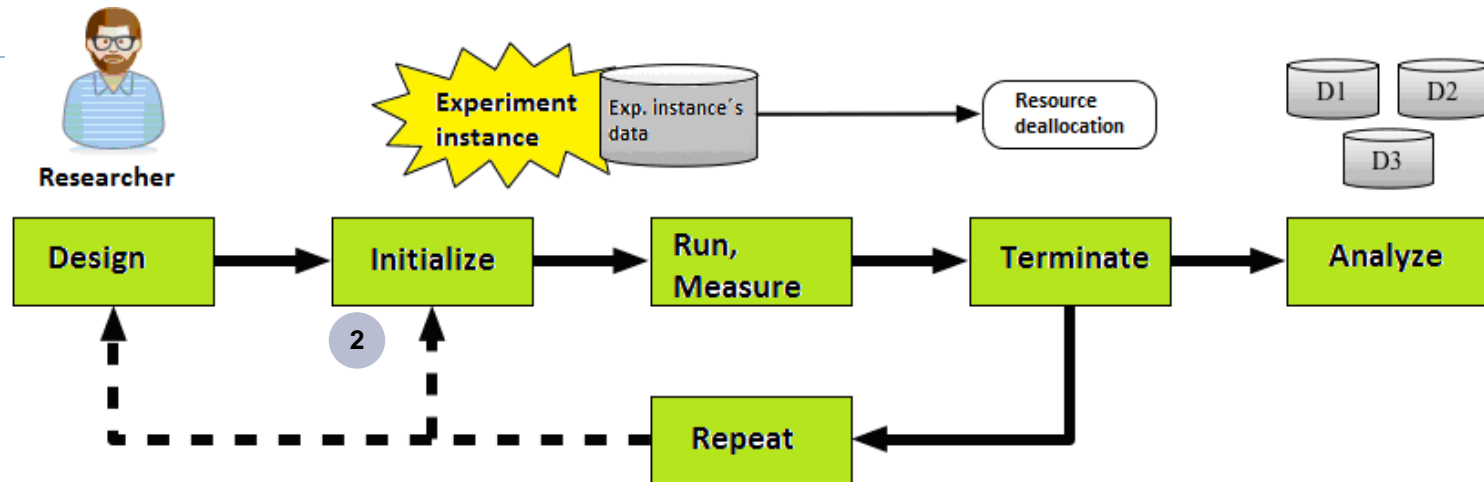


Como funciona?



- 1 Usuário define a topologia desejada em um arquivo ns-2 (diretamente ou através de uma applet Java) e submete o mesmo para o sistema de controle do Emulab

Como funciona?

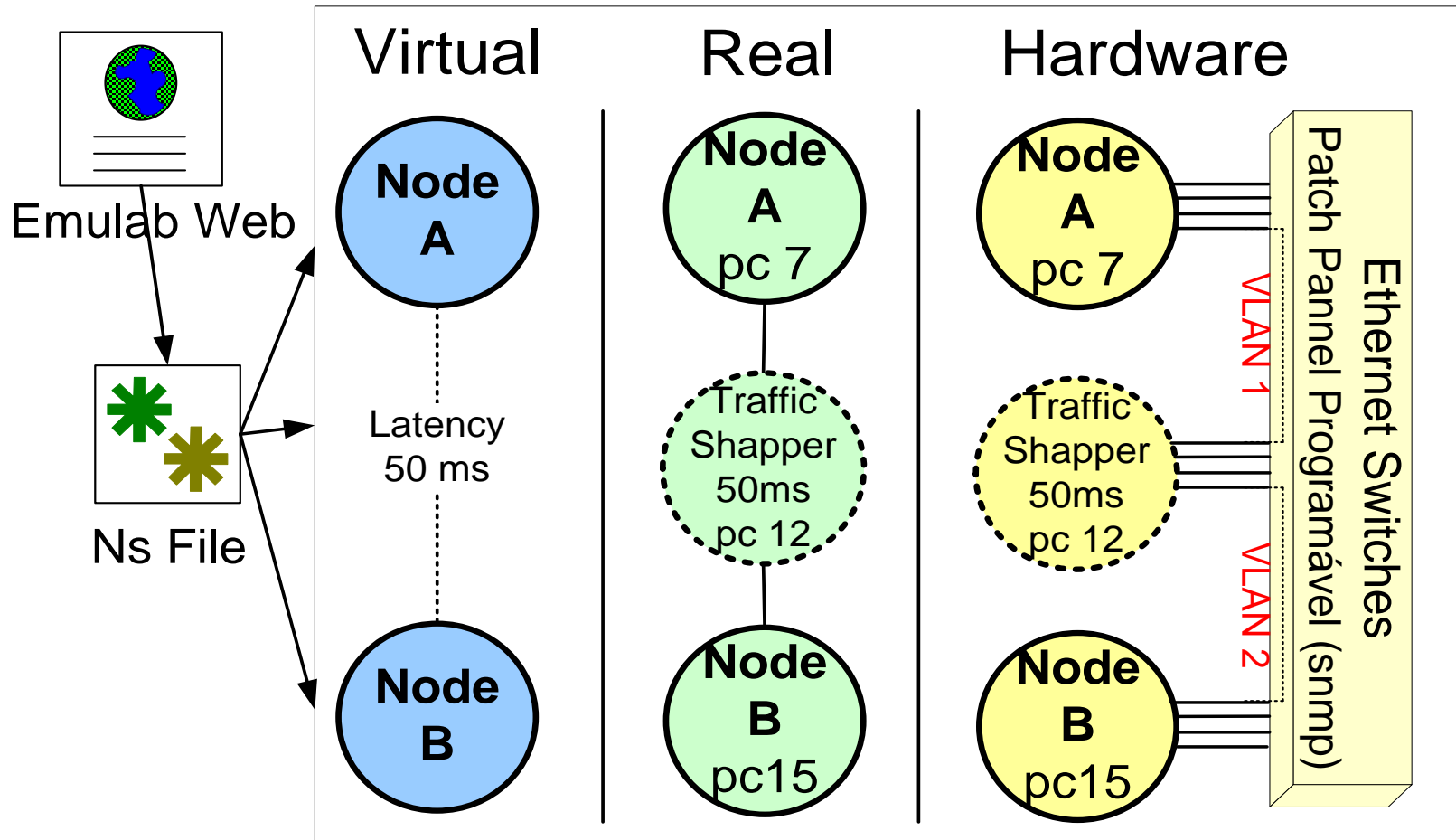


2a Swap-in: o sistema de controle aloca os nós físicos para o experimento

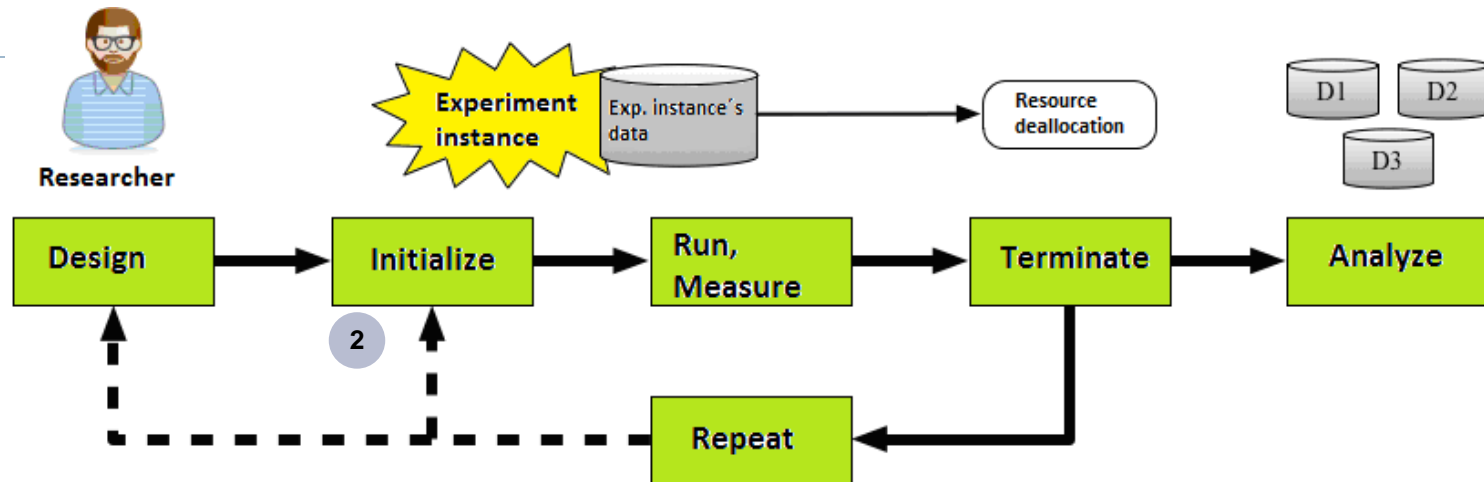
- A topologia virtual é mapeada em recursos físicos, de acordo com o que estiver disponível em um dado momento
- Estes nós são carregados com o SO desejado e configurações básicas
- Acontece a criação de VLANs para os enlaces
- Customização do Experimento
 - Imagens Default / Custom
 - Instalação de Software / execução de Script
 - Definição de rotas estáticas / habilitar roteamento dinâmico

Processo de Criação do Experimento

Experiment



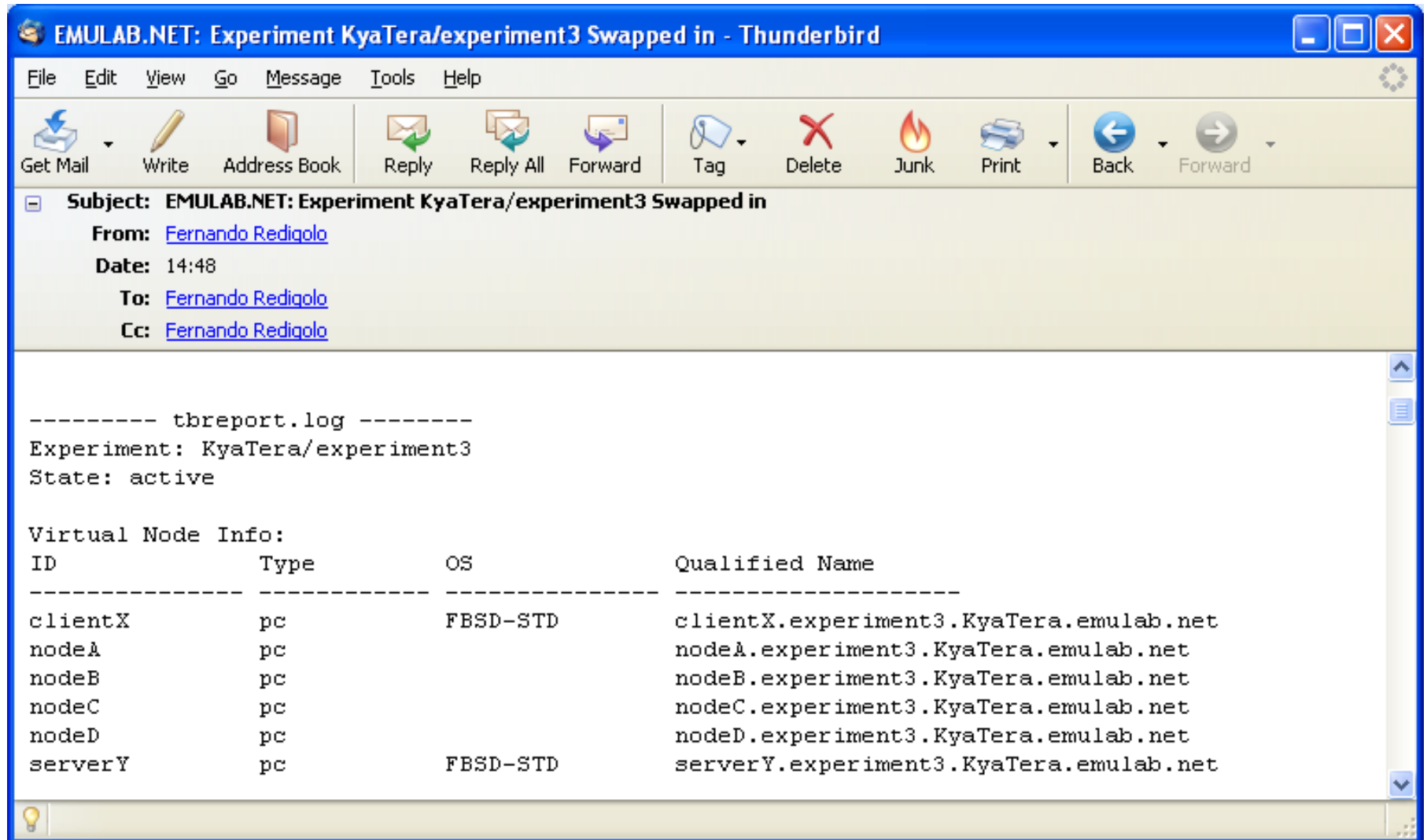
Como funciona?



2b) Sistema envia um e-mail para o usuário notificando que o experimento está pronto para ser usado



Processo de Alocação de Recursos



EMULAB.NET: Experiment KyaTera/experiment3 Swapped in - Thunderbird

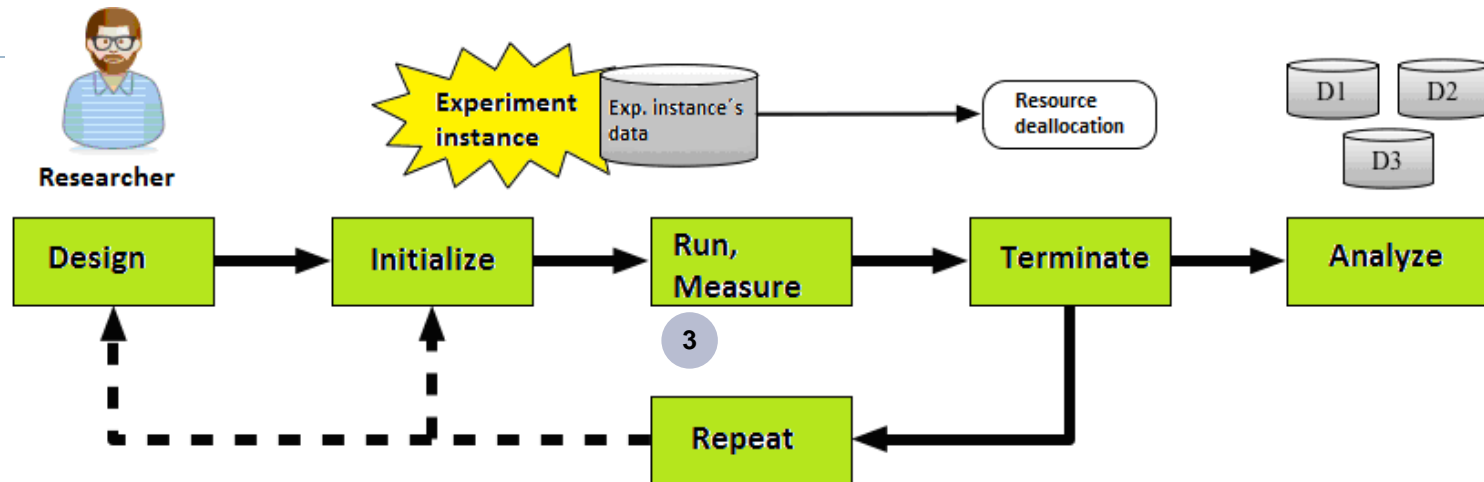
File Edit View Go Message Tools Help

Get Mail Write Address Book Reply Reply All Forward Tag Delete Junk Print Back Forward

Subject: EMULAB.NET: Experiment KyaTera/experiment3 Swapped in
From: [Fernando Redigolo](#)
Date: 14:48
To: [Fernando Redigolo](#)
Cc: [Fernando Redigolo](#)

```
----- tbreport.log -----  
Experiment: KyaTera/experiment3  
State: active  
  
Virtual Node Info:  
ID                Type      OS                Qualified Name  
-----  
clientX           pc       FBSD-STD         clientX.experiment3.KyaTera.emulab.net  
nodeA             pc       FBSD-STD         nodeA.experiment3.KyaTera.emulab.net  
nodeB             pc       FBSD-STD         nodeB.experiment3.KyaTera.emulab.net  
nodeC             pc       FBSD-STD         nodeC.experiment3.KyaTera.emulab.net  
nodeD             pc       FBSD-STD         nodeD.experiment3.KyaTera.emulab.net  
serverY           pc       FBSD-STD         serverY.experiment3.KyaTera.emulab.net
```

Como funciona?



3 Recursos acessíveis remotamente

Tipicamente através de ssh ou console serial

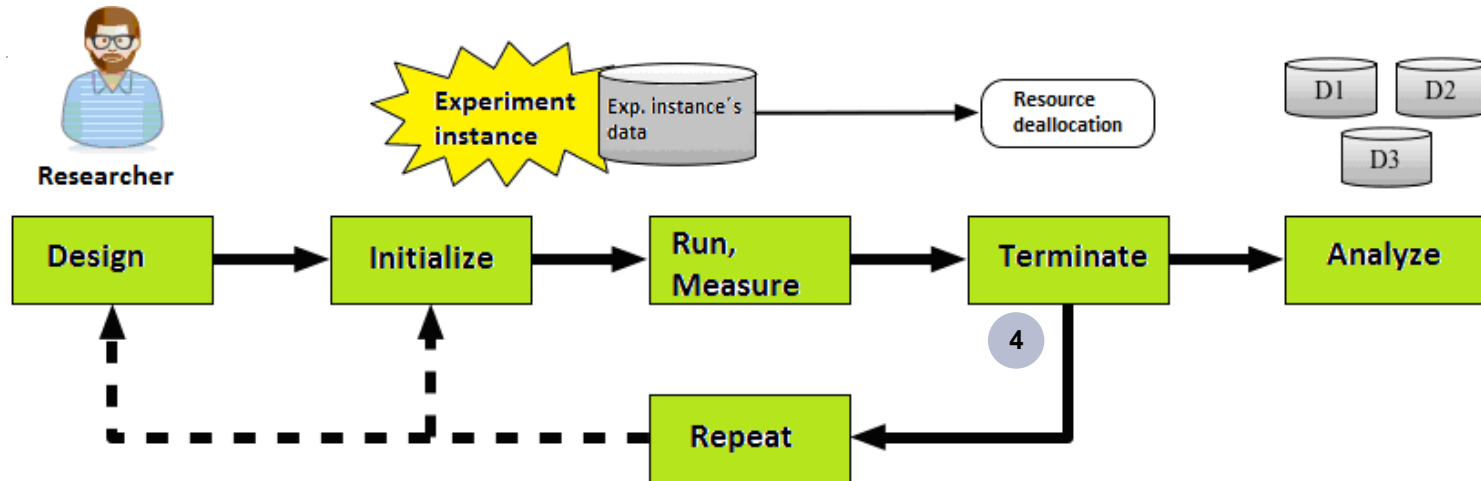
```
EMULAB.NET: Experiment KyaTera/experiment3 Swapped in - Thunderbird
File Edit View Go Message Tools Help
Get Mail Write Address Book Reply Reply All Forward Tag Delete Junk Print Back Forward
Subject: EMULAB.NET: Experiment KyaTera/experiment3 Swapped in
From: Fernando Rediqolo
Date: 14:48
To: Fernando Rediqolo
Cc: Fernando Rediqolo

----- tbreport.log -----
Experiment: KyaTera/experiment3
State: active

Virtual Node Info:
ID          Type      OS          Qualified Name
-----
clientX     pc        FBSD-STD    clientX.experiment3.KyaTera.emulab.net
nodeA       pc        FBSD-STD    nodeA.experiment3.KyaTera.emulab.net
nodeB       pc        FBSD-STD    nodeB.experiment3.KyaTera.emulab.net
nodeC       pc        FBSD-STD    nodeC.experiment3.KyaTera.emulab.net
nodeD       pc        FBSD-STD    nodeD.experiment3.KyaTera.emulab.net
serverY     pc        FBSD-STD    serverY.experiment3.KyaTera.emulab.net
```

Endereços dos nós com SSH

Como funciona?



4 Swap-out: recursos são liberados para outros experimentos

- Processo de liberação de recursos alocados
- 2 principais motivos para um swap-out automático
 - Idle-Swap: número de horas que os recursos não tenham sido usados (ex.: 2hs)
 - Max-duration: número de horas que os recursos podem ficar alocados (ex.: 16hs)
 - Normalmente estes limites são definidos durante a criação do experimento
- Manual swap-out: o pesquisador libera os recursos

Agenda

- ▶ **Parte I - Emulab**
 - ▶ **Emulab LARC/USP**
 - ▶ **Como funciona o Emulab**
- ▶ **Parte II – ProtoGENI**
 - ▶ O que é o ProtoGENI
 - ▶ Arcabouço de Controle e Monitoramento ProtoGENI
 - ▶ *Backbone* ProtoGENI
 - ▶ Federação ProtoGENI
- ▶ **Parte III – Demonstração**

O que é o ProtoGENI?

- ▶ ProtoGENI foi criado para prover a integração de *testbeds* Emulab para poder construir uma infraestrutura GENI
- ▶ A integração consiste de 3 elementos principais:
 - ▶ CMF: *Software* de Arcabouço de Controle e Monitoramento da Universidade de Utah, baseado em uma versão melhorada do *software* de gerenciamento da *testbed* Emulab
 - ▶ Backbone: Uma infraestrutura baseada em *wave* da Internet2 dedicada e de alta velocidade interligando a nível nacional EUA
 - ▶ Federação: Federar com um conjunto de *testbeds*, incluindo uma variedade dessas, como *testbed* em redes sem fio (CMULab), *testbed* de banda larga residencial (CMULab), e os *clusters* programáveis nas bordas (Emulab) que estarão conectados ao backbone

Arquitetura SFA (revisão)

- ▶ A **Clearing House** se comporta como um elemento de coordenação central de registro da federação
- ▶ Cada Gerenciador de Agregados (**Component Manager**) é responsável pela alocação, operação e controle de um ou mais recursos
- ▶ Autoridades de Fatias de Experimentação (**Slice Authorities**) são responsáveis pela coordenação, contabilidade e mediação dos recursos providos pelos vários gerenciadores de agregados (aggregate managers)
- ▶ O **RSpec** é um formato de troca de dados descrevendo as propriedades dos recursos. Ele é usado principalmente em quatro grandes contextos: anúncios (*advertisements*), requisições (*requests*), tickets, e manifestos (*manifests*).

ProtoGENI CMF

- ▶ **Versão melhorada do Arcabouço de Controle Emulab:**
 - ▶ Arquitetura baseada em SFA: Aggregate Manager (AM), Clearing House (CH), Slice Authority (SA), RSpec;
 - ▶ Interface XMLRPC entre AM, CM, SA;
 - ▶ I&M Tools: INSTOOLS e LAMP (baseados em perfSONAR);
 - ▶ Interface Gráfica Flack

Flack

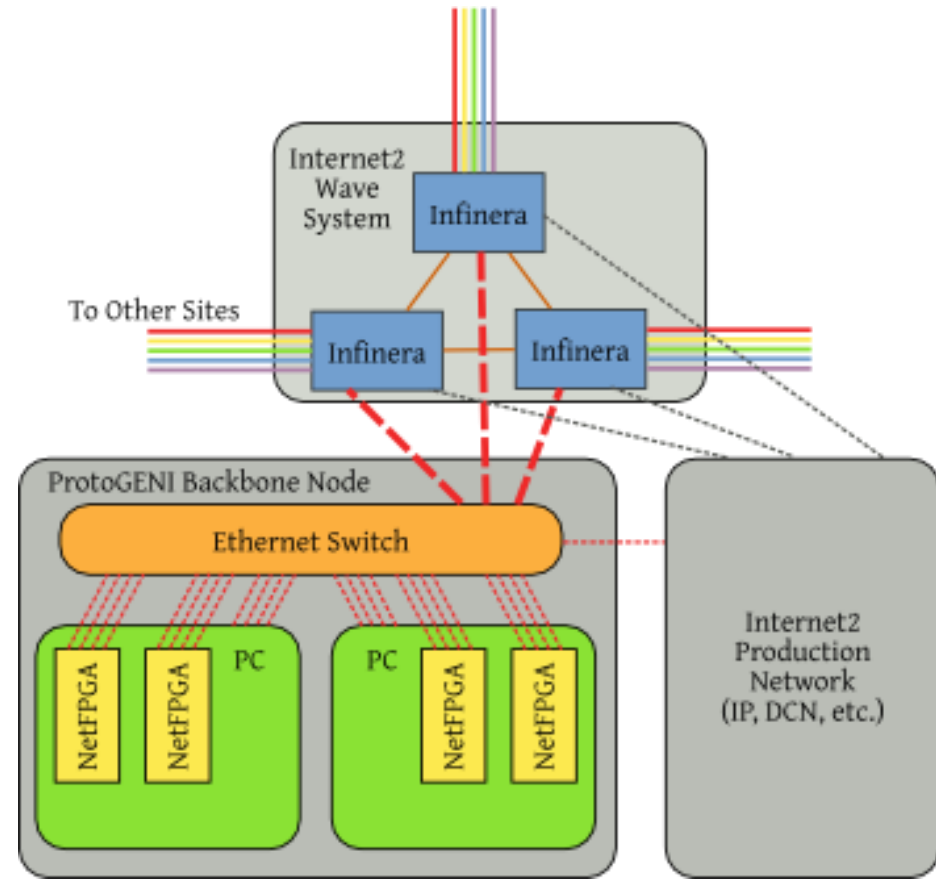
- ▶ Flack é um cliente Google Map baseado em Flash que permite visualização e manipulação de recursos federados



Backbone ProtoGENI

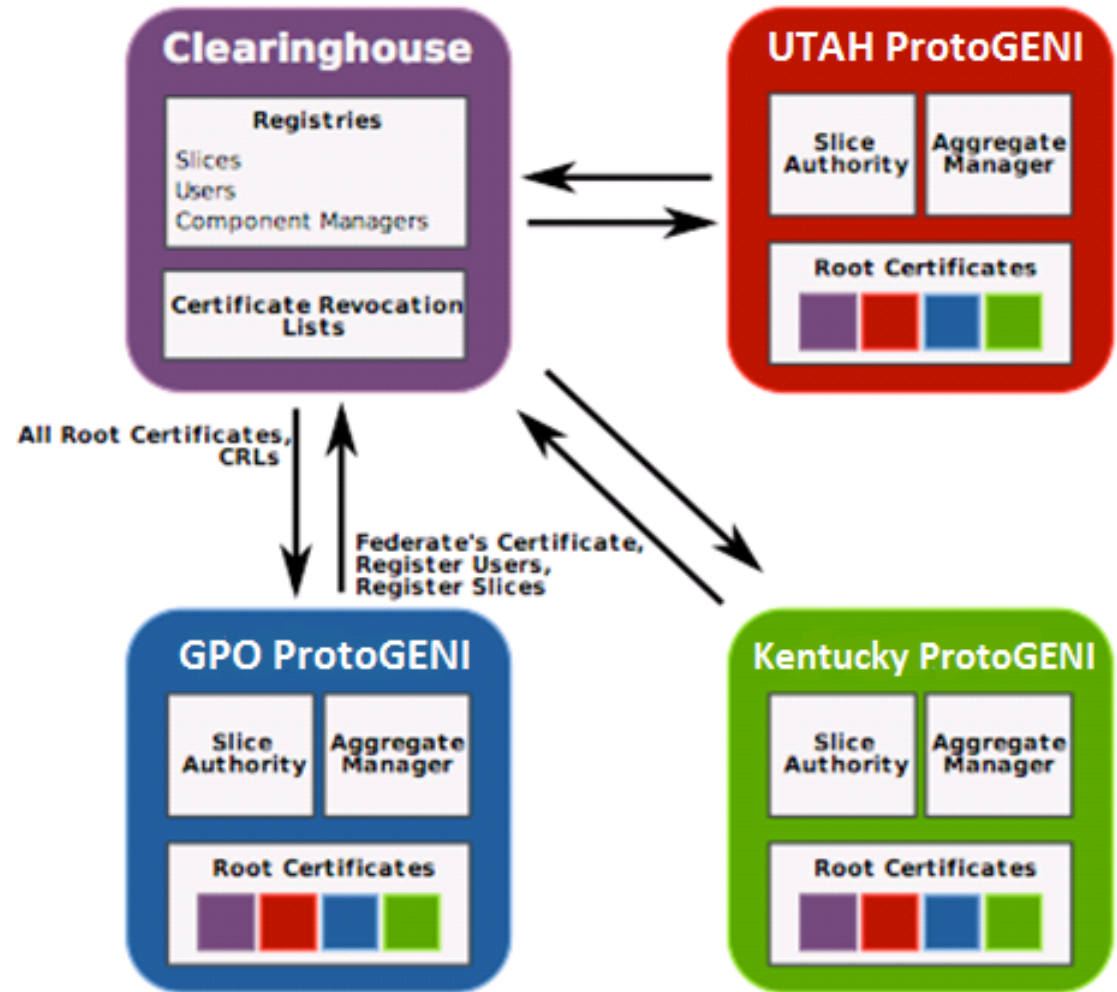
- ▶ Composto de 8 nós conectados por waves de 1 Gbps na infraestrutura da Internet2
- ▶ O *backbone* é baseado em Ethernet nestas waves, e é fatiado com VLANs;
- ▶ Cada nó contém um conjunto de componentes fatiáveis (slices) e programáveis pelos usuários:
 - ▶ PCs com placas NetFPGA
 - ▶ Switches Ethernet (HP Procurve) com capacidade de OpenFlow em implementação

Exemplo de Nó Backbone

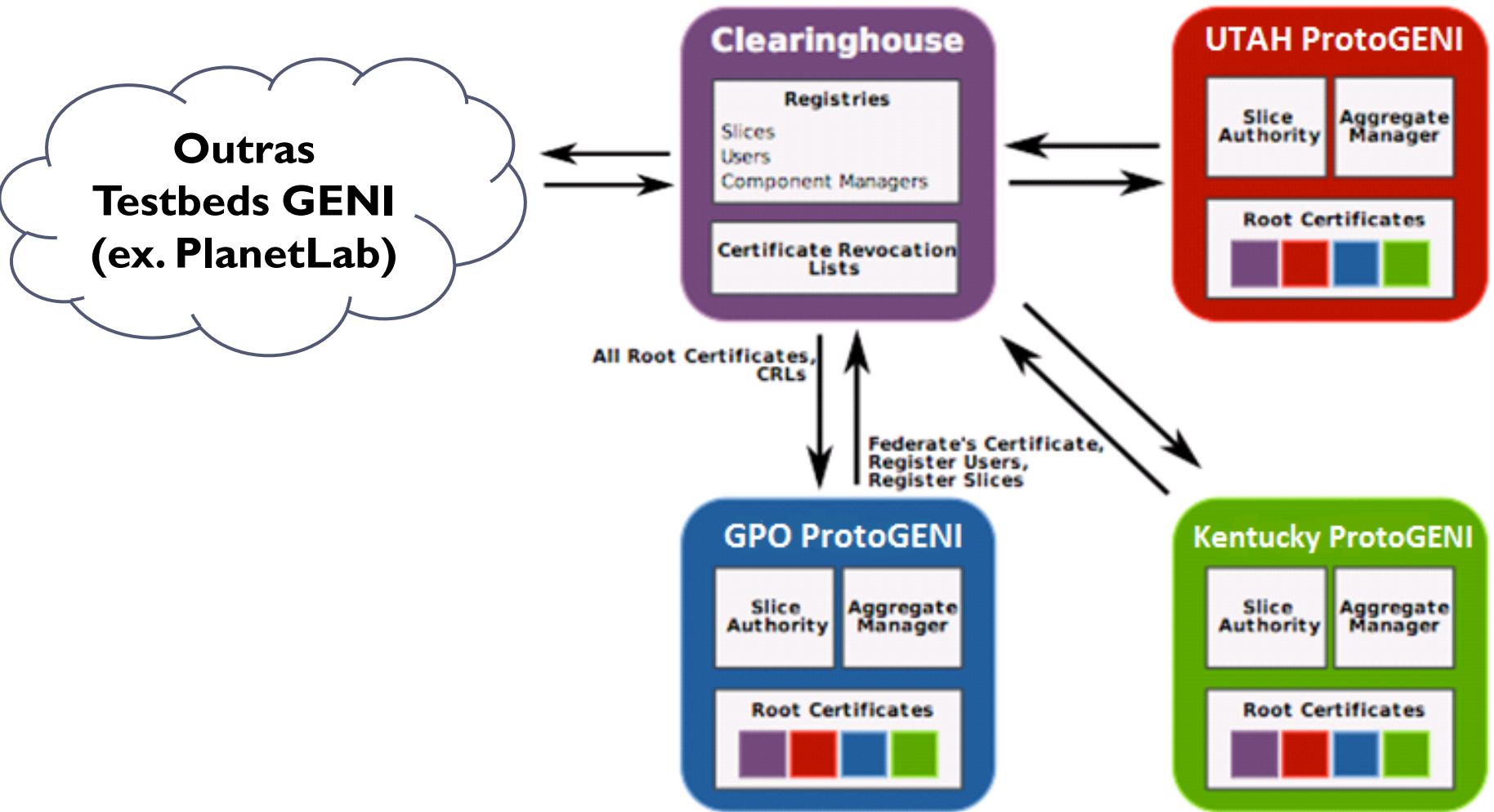


Federação ProtoGENI

- ▶ Integração de *Testbeds*



Federação GENI



Agenda

- ▶ **Parte I - Emulab**
 - ▶ Emulab LARC/USP
 - ▶ Como funciona o Emulab
- ▶ **Parte II – ProtoGENI**
 - ▶ O que é o ProtoGENI
 - ▶ Arcabouço de Controle e Monitoramento ProtoGENI
 - ▶ *Backbone* ProtoGENI
 - ▶ Federação ProtoGENI
- ▶ **Parte III – Demonstração**

Demonstração

▶ Pré-requisitos

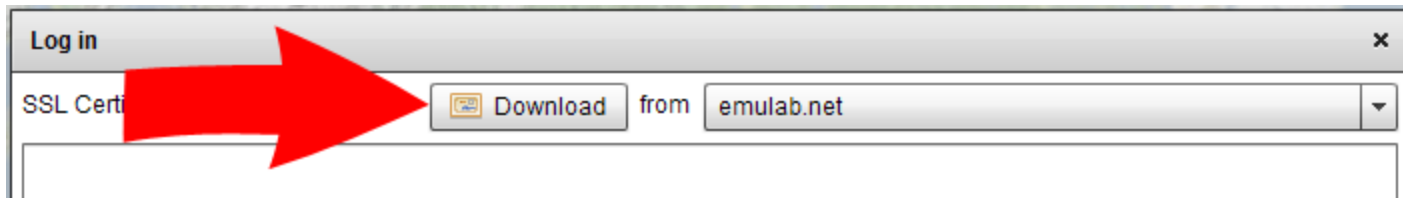
- ▶ É preciso ter uma conta em um dos gerenciadores de agregados do Emulab
- ▶ A conta pode ser solicitada no link: www.emulab.net, clicando no botão "*Request Account*"
- ▶ Observar que a criação de conta pode requerer aprovação e isso pode durar um certo tempo

Acessando a Ferramenta Flack

- ▶ Primeiramente, realizar o login Flack no link: <http://protogeni.net/flack>.



- ▶ Flack usa as credenciais geradas de uma autoridade certificadora. Selecione essa autoridade e realize o download das credenciais

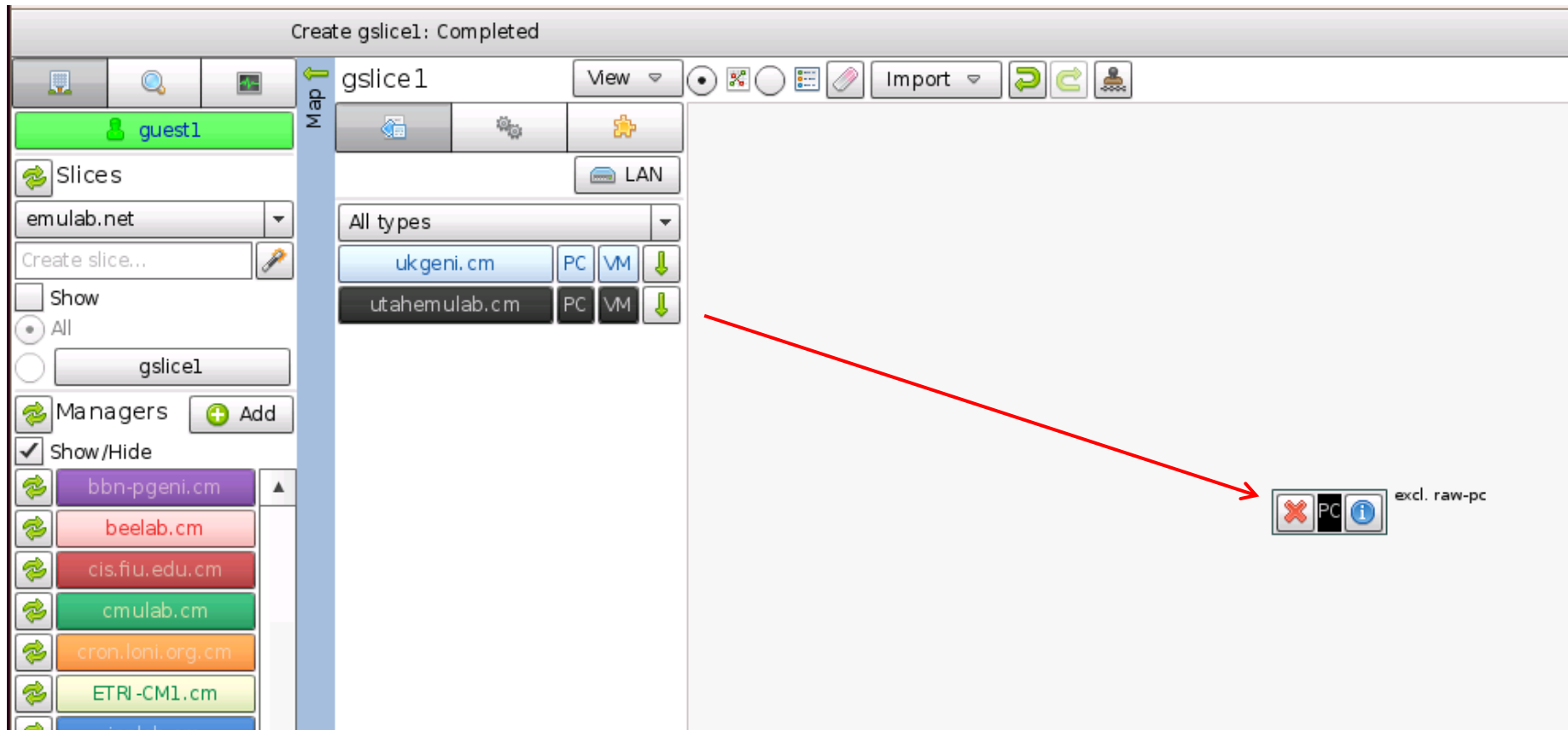


Criando uma Fatia de Experimentação



Definição do Experimento

- ▶ Nós podem ser adicionados à “fatia” arrastando-os do gerenciador de agregados de onde voce quer criar o nó



Definição do Experimento

- ▶ Nós podem ser adicionados à “fatia” arrastando-os do gerenciador de agregados de onde voce quer criar o nó

Create gslice1: Completed

Map gslice1 View Import

LAN

All types

ukgeni.cm	PC	VM	↓
utahemulab.cm	PC	VM	↓

excl. raw-pc

Tipos comuns de nós:

- PC (máquina física com acesso exclusivo);
- VM (máquina virtual compartilhada).

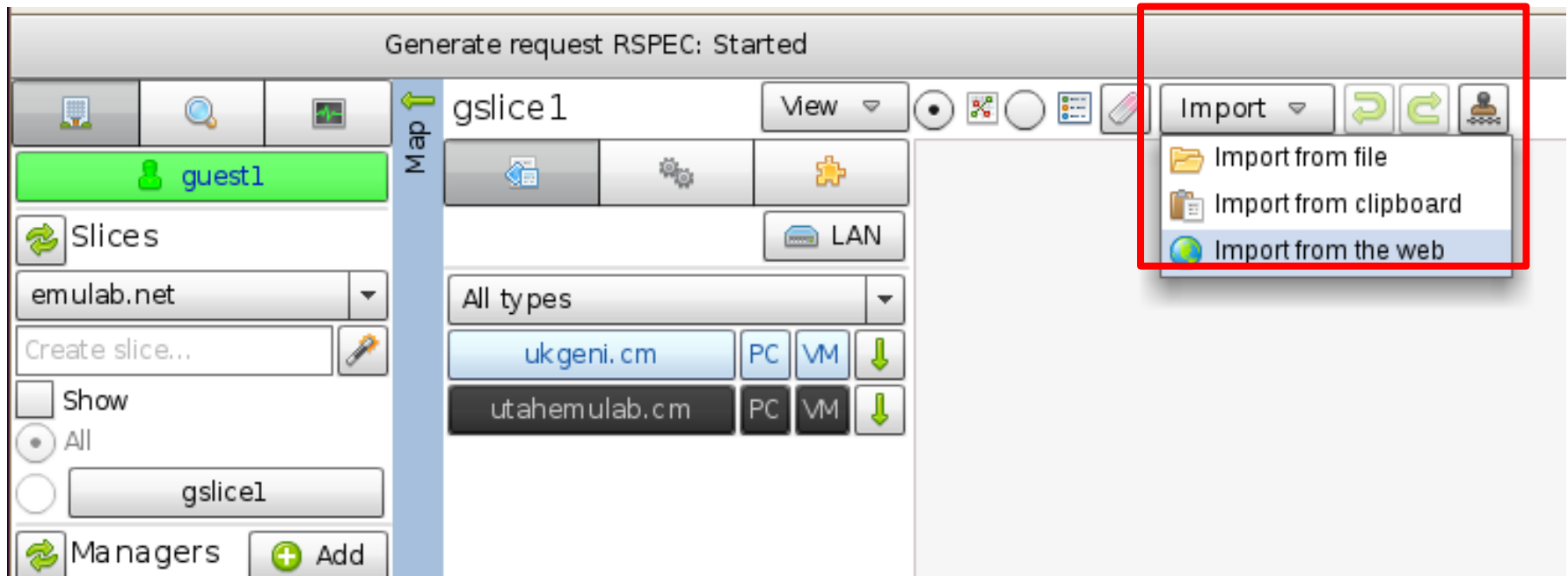
Definição de Experimento

- ▶ Enlaces entre os nós são criados por somente arrastando uma linha de um nó para outro

The screenshot displays the 'gslice1' network configuration interface. The top bar indicates 'Generate request RSPEC: Started'. The interface includes a toolbar with icons for home, search, and map, and a 'View' dropdown. Below the toolbar, there are buttons for 'Import', a refresh icon, and a user icon. The left sidebar contains a 'Map' section with a 'quest1' user profile, a 'Slices' section with 'emulab.net' and a 'Create slice...' button, and a 'Managers' section with a list of managers including 'bbn-pgeni.cm', 'beelab.cm', 'cis.fiu.edu.cm', 'cmulab.cm', 'cron.loni.org.cm', 'ETRI-CM1.cm', 'jonlab.cm', and 'mygeni.cm'. The main area shows a network diagram with nodes: 'PC-0' (excl. raw-pc), 'lan0', 'PC' (excl. raw-pc), and 'PC-1' (excl. raw-pc). Lines connect 'PC-0' to 'lan0' and 'lan0' to 'PC'. A green arrow points to the 'PC-1' node.

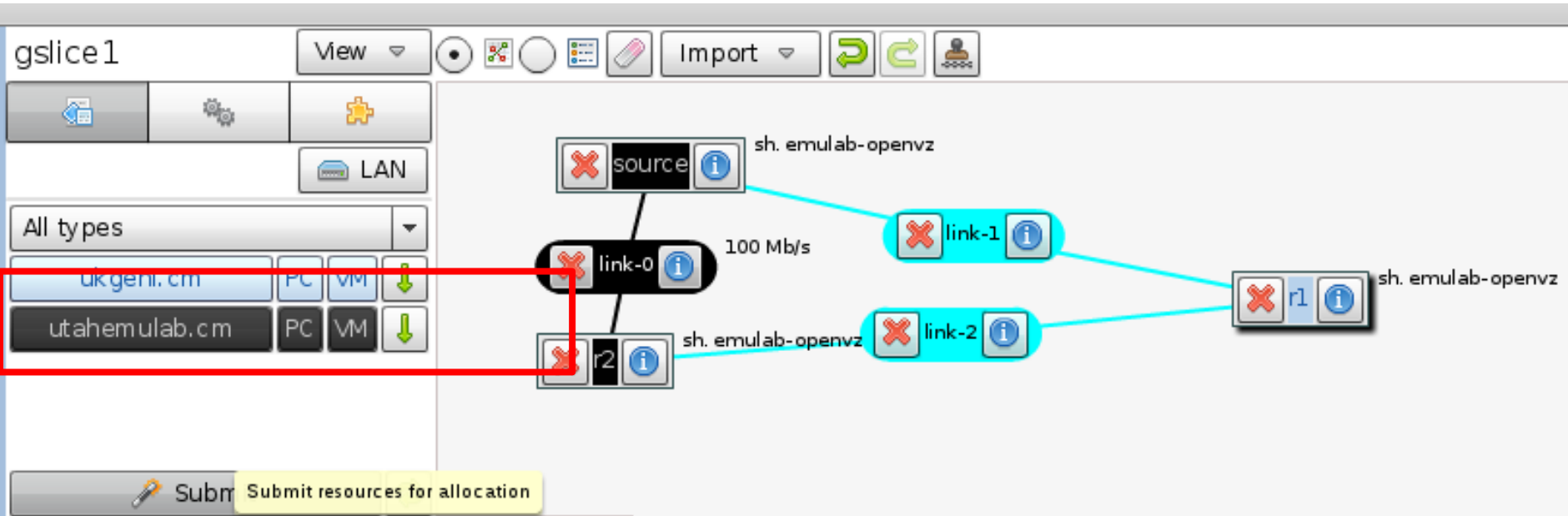
Definição do Experimento

- ▶ Opcionalmente carrega uma descrição RSpec do experimento



Criação da Fatia de Experimentação

- ▶ Quando a topologia estiver pronta, a fatia poderá ser criada.



- ▶ O acesso aos nós é feito por ssh (como no Emulab)

CMF: OMF Detalhamento

Kleber Cardoso (UFG)



Agenda

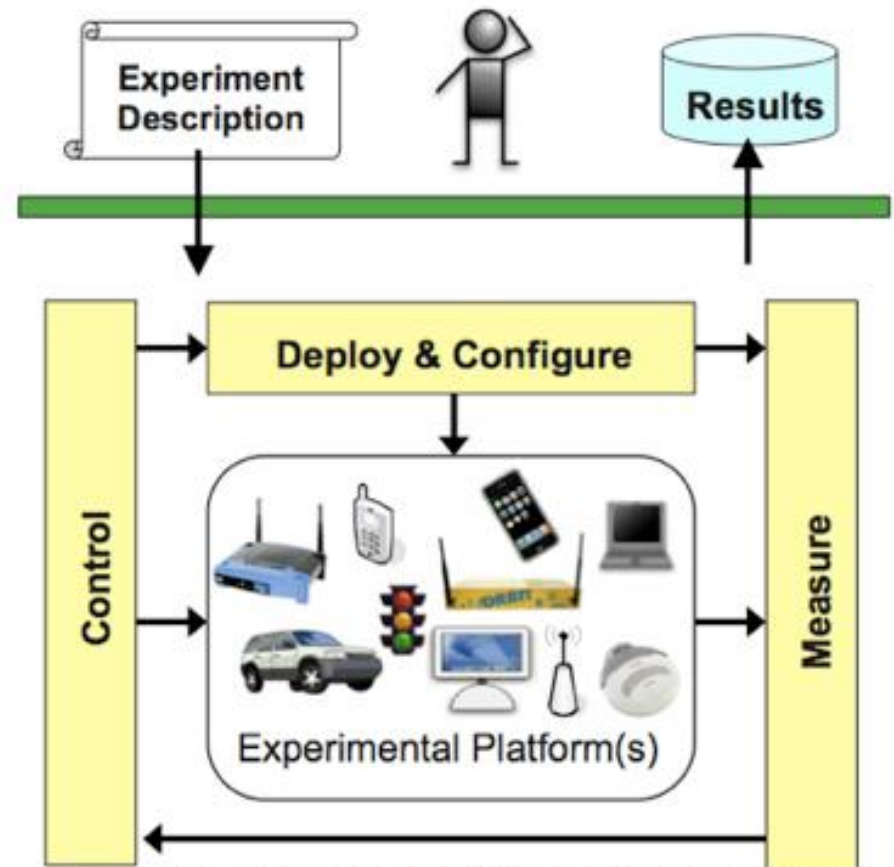
- ▶ Visão geral
- ▶ Arquitetura
- ▶ Exemplos
- ▶ Instalação
 - ▶ Requisitos e infraestrutura
 - ▶ Configuração da rede
 - ▶ Serviços
 - ▶ Testes
- ▶ Demonstração em Vídeo

Visão geral

- ▶ Projeto iniciado em 2003, na Universidade Rutgers (no Winlab) em conjunto com o NICTA (National ICT Australia)
- ▶ Proposto originalmente para redes sem fio
 - ▶ Originalmente, OMF expandia para *ORBIT Management Framework*
 - ▶ ORBIT expande para **O**pen-Access **R**esearch Test**B**ed for Next-Generation **W**ireless Ne**T**works
- ▶ Atualmente, suporta outras tecnologias
 - ▶ OMF passou a expandir para **c**Ontrol and **M**anagement **F**ramework
 - ▶ A parte de medições (*OML – OMF Measurement Library*) passou a ser tratada como um recurso independente
 - ▶ Desde 2008, o código oficial passou a ser mantido no site mytestbed.net, recebendo colaborações do Winlab/Rutgers, NICTA, Nitlab/Thessaly, dentre outros

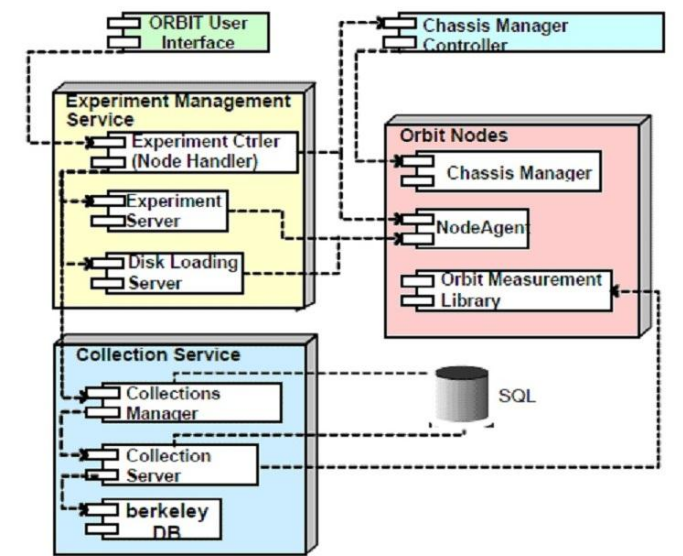
•Visão geral (2)

- ▶ OMF sob o ponto de vista do experimentador:
 - ▶ “conjunto de ferramentas para descrever e instrumentar um experimento, executá-lo e coletar seus resultados”
- ▶ OMF sob o ponto de vista do operador da testbed:
 - ▶ “conjunto de serviços para gerenciar e operar os recursos da testbed (ex.: reinicializar dispositivos, obter informações de status, instalar imagem de SO)”

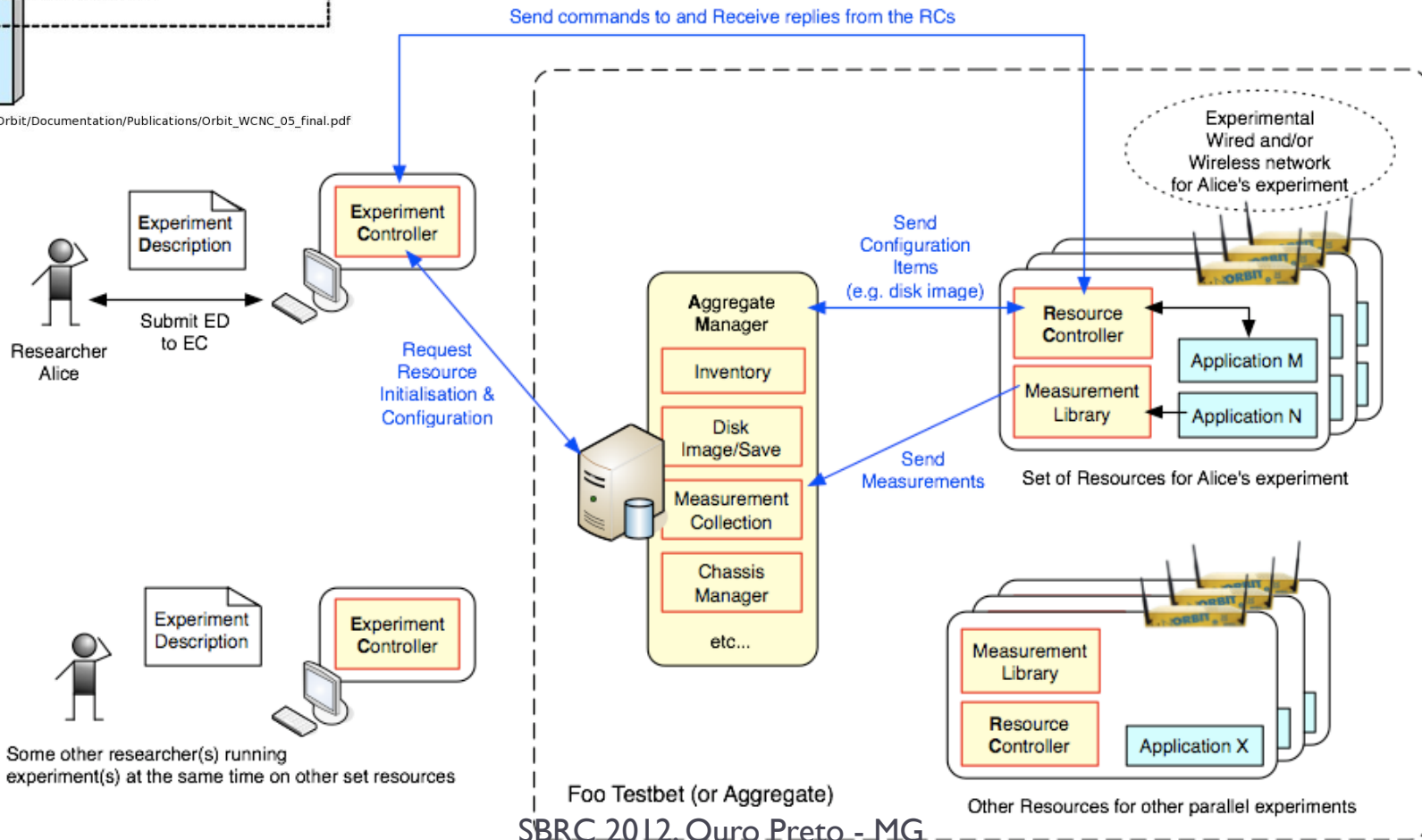


Fonte: <http://mytestbed.net/projects/omf/introduction>

Arquitetura



http://www.orbit-lab.org/attachment/wiki/Orbit/Documentation/Publications/Orbit_WCNC_05_final.pdf

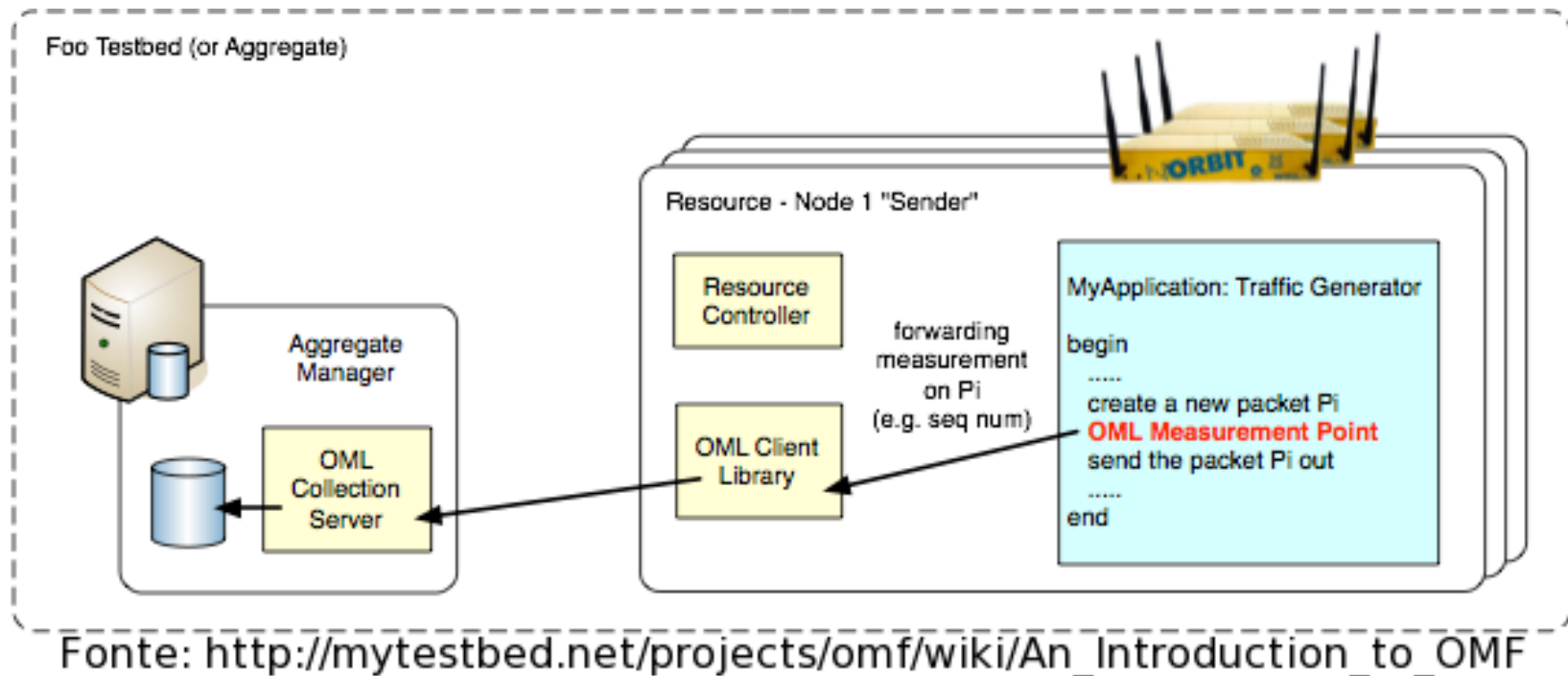


SBRC 2012, Ouro Preto - MG

Fonte: http://mytestbed.net/projects/omf/wiki/An_Introduction_to_OMF

Arquitetura – medições

- ▶ OML pode ser usada em conjunto com o OMF ou de maneira independente



Exemplos

- ▶ **Exemplos de pesquisas em testbeds com OMF**
 - ▶ Redes veiculares, redes em malhas, nós móveis
 - ▶ Vários tipos de avaliação com 802.11: controle de taxa, troca de canais, efeito de captura, etc.
 - ▶ WiMax, dentre outros
- ▶ **Algumas testbeds com OMF:**
 - ▶ ORBIT (Winlab, Rutgers): várias testbeds pequenas (2 a 8 com diferentes recursos como 802.11, SDR, atenuador de RF, estação base WiMax) e uma grande (com 400 nós)
 - ▶ NORBIT (NICTA, Austrália): 5 testbeds, variando de 3 a 34 nós, sendo 1 externa e 4 internas
 - ▶ NITOS (NitLab, Grécia): 35 nós espalhados por um prédio, incluindo interfaces 802.11 e SDR
 - ▶ Witest Lab (Polytechnic institute of NYU, EUA): uma testbed interna com 15 nós 802.11 e uma estação base WiMax

Requisitos e infraestrutura

- ▶ **Componentes críticos:**
 - ▶ gerenciador de agregado (AM – aggregate manager)
 - ▶ controlador de experimentos (EC – experiment controller) – pode estar fora da testbed
 - ▶ controlador de recurso (RC – resource controller)
- ▶ **Componente opcional, mas muito importante:**
 - ▶ OML, o qual está dividido em três partes: servidor, cliente e proxy (opcional)

Requisitos e infraestrutura (2)

- ▶ **Hardware e software básico:**
 - ▶ Teoricamente, qualquer arquitetura e SO
 - ▶ Recomendado: arquitetura x86 e SO Linux
 - ▶ Será comentada a instalação sobre Ubuntu 11.10 “oneiric” porque há pacotes binários para todos os componentes
- ▶ **Ferramentas básicas:**
 - ▶ Interpretador Ruby e servidor XMPP
 - ▶ No entanto, o OMF não é operacional sem outros recursos “auxiliares” como tradução de nomes e banco de dados

Requisitos e infraestrutura (3)

- ▶ Testbed mínima:
 - ▶ 1 “servidor”
 - ▶ 1 PC atual (ex.: Intel i3, 2 GB de RAM, 50 GB de disco)
 - ▶ É útil ter duas interfaces de rede Ethernet
- ▶ 2 nós
 - ▶ Qualquer hardware que execute um interpretador Ruby
 - ▶ Interfaces de comunicação para testes (ex.: 802.11)
 - ▶ É útil ter duas interfaces de rede Ethernet
 - ▶ Se for um hardware do tipo PC, é possível:
 - ▶ Trocar imagem do SO remotamente
 - ▶ Usar aplicações convencionais

.Configuração da Rede

▶ No servidor:

▶ Desabilitar configuração automática. Exemplo:

```
apt-get remove network-manager
```

▶ Configuração manual. Exemplo:

```
auto eth0
iface eth0 inet dhcp
auto eth1
iface eth1 inet static
address 10.0.0.200
netmask 255.255.255.0
```

▶ Serviços SSH e NTP. Exemplo:

```
apt-get install ssh ntp
```

Serviços

- ▶ Configuração automática de endereços IP – é utilizada para configurar a interface de controle dos nós de teste. Software: **dnsmasq**
- ▶ Tradução de nomes – permite que nomes, ao invés de endereços IP, sejam usados para identificar os servidores (AM e OML) e os próprios nós de experimentação. Software: **dnsmasq**
- ▶ Inicialização remota – auxilia na customização completa de um nó, permitindo que até o sistema operacional seja substituído na inicialização do equipamento. Software: **dnsmasq** + **Syslinux**
- ▶ *Middleware* orientado a mensagens – constitui o núcleo do OMF, provendo a comunicação entre AM, EC, RC e OML para fornecer as funcionalidade de controle e monitoramento dos experimentos. Software: **Openfire**
- ▶ Base de dados – armazena o inventário dos nós, usuários, resultados coletados e demais informações para operação e uso da *testbed*. Software: **MySQL** e **SQLite** (OML)

Serviços – dnsmasq

▶ Instalação. Exemplo:

```
apt-get install syslinux dnsmasq
```

▶ Configuração básica (/etc/dnsmasq.conf). Exemplo:

```
interface=eth1
dhcp-range=10.0.0.201,10.0.0.254,255.255.255.0,12h
dhcp-option=3
dhcp-option=option:ntp-server,10.0.0.200
dhcp-boot=net:control,pxelinux.0
enable-tftp
tftp-root=/tftpboot
```

▶ Configuração automática das interfaces de controle (/etc/dnsmasq.d/mytestbed.conf). Exemplo:

```
dhcp-host=net:control,00:01:02:03:04:05,node1,10.0.0.1
dhcp-host=net:control,00:06:07:08:09:10,node2,10.0.0.2
dhcp-host=net:control,00:AA:BB:CC:DD:EE,node3,10.0.0.3
```

Serviços – Syslinux

- ▶ Criar os diretórios que conterão os arquivos de configuração e inicialização usados para o boot remoto PXE e referência ao carregador de inicialização. Exemplo:

```
mkdir -p /tftpboot/pxelinux.cfg  
ln -s /usr/lib/syslinux/pxelinux.0 /tftpboot/
```

- ▶ Arquivo de configuração específico (/tftpboot/pxelinux.cfg/pxeconfig). Exemplo:

```
SERIAL 0 19200 0  
DEFAULT linux  
LABEL linux  
KERNEL linux-omf-pxe-3.0.4  
APPEND console=tty0 console=ttyS0,19200n8 vga=normal quiet  
root=/dev/ram0 rw load_ramdisk=1 prompt_ramdisk=1  
ramdisk_size=32768 initrd=initramfs-omf-pxe-5.4.bz2  
control=eth0 xmpp=srv.fibre.ufg.br slice=pxe_slice  
hrn=omf.fibre.%hostname%  
PROMPT 0
```

Serviços – Syslinux (2)

- ▶ Criar arquivo configuração padrão (/tftpboot/pxelinux.cfg/default). Exemplo:

```
DEFAULT harddrive  
LABEL harddrive  
localboot 0
```

- ▶ Apenas para testes, deve ser criado um link simbólico para cada nó

```
ln -s pxeconfig 01-00-01-02-03-04-05  
ln -s pxeconfig 01-00-06-07-08-09-10  
ln -s pxeconfig 01-00-AA-BB-CC-DD-EE
```

Serviços – Openfire

- ▶ Para a versão 5.4 do OMF, é recomendado instalar o Openfire 3.7.1 e o SUN Java 6. Exemplo:

```
apt-get install python-software-properties
add-apt-repository ppa:ferramroberto/java
apt-get update; apt-get install sun-java6-jre
wget http://www.igniterealtime.org/downloadServlet?filename=
openfire/openfire_3.7.1_all.deb -O openfire_3.7.1_all.deb
dpkg -i openfire_3.7.1_all.deb
```

- ▶ Verificar se as portas TCP 5222, 5269 e 9090 estão acessíveis
- ▶ Configuração do Openfire é feita através de uma interface Web que pode ser acessada através da porta 9090 do servidor

AM e servidor OML

- ▶ Inclusão do site de pacotes do projeto OMF (/etc/apt/sources.list). Exemplo:

```
deb http://pkg.mytestbed.net/ubuntu oneiric/
```

- ▶ Instalação do AM e do servidor OML. Exemplo:

```
apt-get update  
apt-get install omf-aggmgr-5.4 oml2-server
```

AM e servidor OML (2)

- ▶ Copiar `/usr/share/doc/omf-aggmgr-5.4/examples/omf-aggmgr.yaml` para `/etc/omf-aggmgr-5.4/`
- ▶ Alterar o arquivo `/etc/omf-aggmgr-5.4/omf-aggmgr.yaml` para indicar o nome correto do servidor XMPP
- ▶ Ativar os serviços do OMF através da criação dos seguintes links simbólicos:

```
cd /etc/omf-aggmgr-5.4/enabled; ln -s ../available/cmcStub.yaml;  
ln -s ../available/frisbee.yaml; ln -s ../available/pxe.yaml;  
ln -s ../available/inventory.yaml; ln -s ../available/result.yaml;  
ln -s ../available/saveimage.yaml
```

- ▶ Por fim, dentro do diretório `/etc/omf-aggmgr-5.4/available/`, é preciso editar os arquivos `saveimage.yaml` e `frisbee.yaml` para alterar as interfaces para salvar imagens de sistemas operacionais (`saveimage`) e multicast

AM e servidor OML (3)

- ▶ Instalação do servidor MySQL e da ferramenta de gerência com interface Web chamada phpmyadmin. Exemplo:

```
apt-get install mysql-server libdb4.6 phpmyadmin
```

- ▶ Conectar ao servidor MySQL (mysql -u root -p) e criar uma conta para o OMF. Exemplo:

```
use mysql;  
create user 'omf'@'localhost';  
grant all on *.* to 'omf'@'localhost';  
set password for 'omf'@'localhost'=password('omf');  
quit;
```

- ▶ Conexão ao servidor MySQL com o usuário OMF (mysql -u omf -pomf) e criar a base de dados do inventário. Exemplo:

```
create database inventory;  
quit;
```

- ▶ Importar o arquivo SQL de exemplo de inventário. Exemplo:

```
• cd /usr/share/doc/omf-aggmgr-5.4/examples  
zcat inventory.sql.gz | mysql -u omf -pomf inventory
```

AM e servidor OML (4)

- ▶ Criar a árvore de tópicos PubSub. Exemplo:

```
omf_create_psnode-5.4 srv.fibre.ufg.br mksys
```

- ▶ Criar uma fatia de experimentação (slice) com o nome pxe_slice e incluir todos os nós. Exemplo:

```
omf_create_psnode-5.4 srv.fibre.ufg.br mkslice pxe_slice  
omf.fibre.node1 omf.fibre.node2 omf.fibre.node3
```

- ▶ Criação de uma fatia com os nós para realização de testes. Exemplo:

```
omf_create_psnode-5.4 srv.fibre.ufg.br mkslice pxe_slice  
omf.fibre.node1 omf.fibre.node2 omf.fibre.node3
```

- ▶ Configuração concluída, basta reiniciar:

```
/etc/init.d/omf-aggmgr-5.4 restart
```


▶ Instalação do EC. Exemplo:

EC

```
apt-get install omf-expctl-5.4
```

▶ Usar arquivo de exemplo para configuração:

```
cd /usr/share/doc/omf-expctl-5.4/examples/  
zcat omf-expctl.yaml.gz > /etc/omf-expctl-5.4/omf-expctl.yaml
```

▶ Modificar alguns parâmetros do arquivo. Exemplo:

```
:domain: 'fibre_ufg'  
:omluri: 'tcp:srv.fibre.ufg.br:3003'  
:web:  
  :host: '10.0.0.200'  
:communicator:  
  :xmpp:  
    :pubsub_gateway: 'srv.fibre.ufg.br'  
:services:  
  :uri: 'http://srv.fibre.ufg.br:5054'
```

▶ Obter imagem fornecida pelos desenvolvedores:

```
cd /var/lib/omf-images-5.4  
wget http://pkg.mytestbed.net/files/5.4/baseline/baseline.ndz
```

▶ Testar a imagem nos nós:

```
omf load
```

Testes

```
#Define o grupo de recursos 'teste'
defGroup('teste',omf.fibre.node2,omf.fibre.node2)
#Quando todos os nos estiverem prontos,
#execute '/bin/ls -l -a'

onEvent(:ALL_UP_AND_INSTALLED) do |event|
  #Imprime alguma informacao na tela
  puts "Executando '/bin/ls -l -a' nos nos..."
  group("teste").exec("/bin/ls -l -a")

  #Aguarda 10 segundos
  wait 10

  #Encerra o experimento
  Experiment.done
end
```

Testes (2)

```
defGroup('servidor', "omf.fibre.node1") do |node|
  node.net.e1.ip = "192.168.0.1"
  node.addApplication("test:app:iperf") { |app|
    app.setProperty('server', true)
    app.setProperty('port', 5000)
  }
end
```

```
defGroup('cliente', "omf.fibre.node2") do |node|
  node.net.e1.ip = "192.168.0.2"
  node.addApplication("test:app:iperf") do |app|
    app.setProperty('client', '192.168.0.1')
    app.setProperty('port', 5000)
    app.setProperty('time', 30)
    app.setProperty('interval', 1)

    app.measure('TCP_Info', :samples => 1)
  end
end
```

```
onEvent(:ALL_UP_AND_INSTALLED) do |event|
  group("servidor").startApplications
  wait 5
  group("cliente").startApplications
  wait 40
```

```
  Experiment.done
end
```

Demonstração em Vídeo

CMF: OFELIA

Cesar Marcondes (UFSCar)

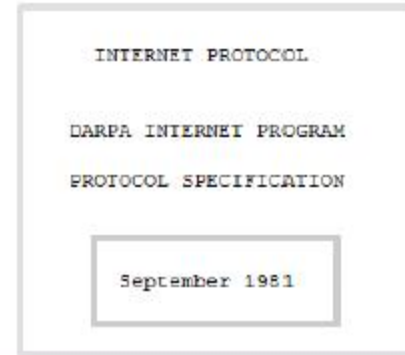


Agenda

- ▶ Breve introdução sobre OpenFlow
- ▶ “Fatias” de Redes com FlowVisor
- ▶ O que é o OFELIA?
- ▶ Arquitetura do OFELIA
- ▶ Utilizando o *testbed* OFELIA
- ▶ Instalação local da *testbed* OFELIA
- ▶ Demonstração em vídeo do uso da *testbed* OFELIA

Ossificação da Internet

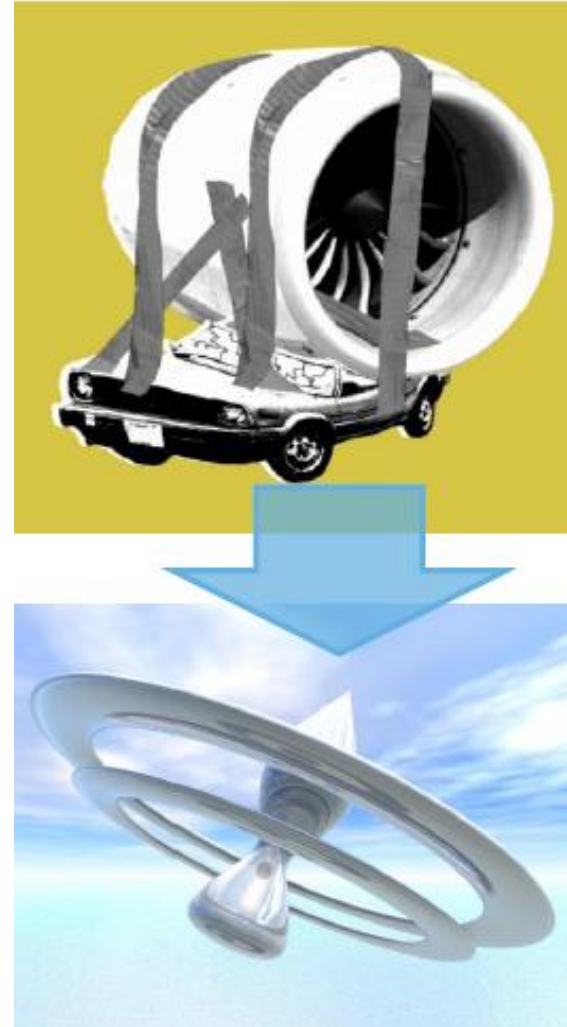
- ▶ O Protocolo IP (IP) forma a base do nosso sistema de comunicação corrente, embora a tecnologia tenha origens nos anos 80
- ▶ muitos remendos foram aplicados de modo que a rede sobrevivesse ao seu próprio sucesso,
- ▶ Funciona ... mas estamos engessados do ponto de vista de inovação



Slides baseados no OFELIA

Inovação versus OpenFlow

- ▶ Inovação em Redes é impossível com sistemas fechados / proprietários
- ▶ É preciso uma solução aberta para implementar novos serviços com curto tempo para mercado
- ▶ Operadores de rede não querem esperar que todos os fabricantes implementem uma inovação antes de ser capaz de lançar um novo serviço
- ▶ A ideia é conseguir de volta o controle de nossas redes, e é onde o *OpenFlow* entra



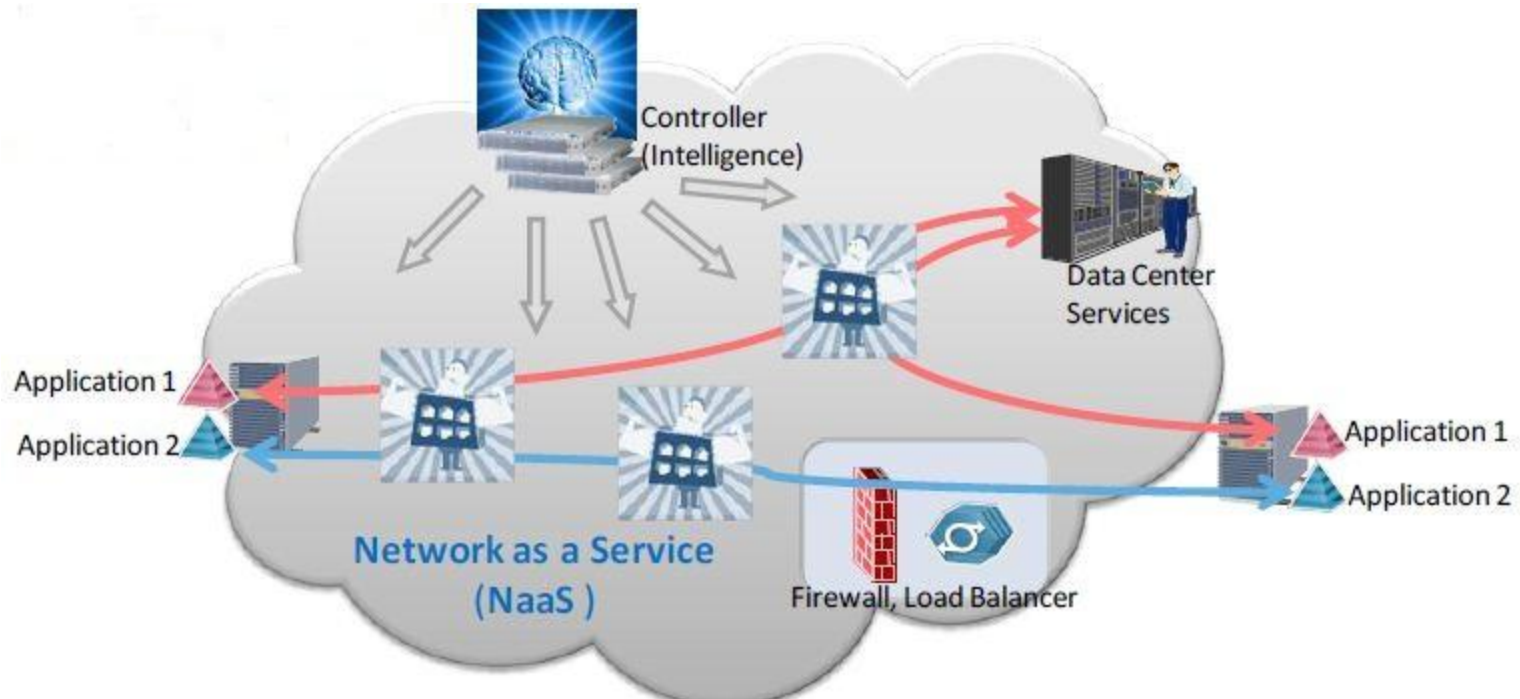
OpenFlow

- ▶ Fruto do Programa de Pesquisa em Internet do Futuro de Stanford
- ▶ Separação do Plano de Controle e de Dados
 - ▶ Uma instancia de um controlador centralizado faz a comunicação com os nós da rede usando o protocolo padrão OpenFlow
 - ▶ As aplicações de rede residem dentro do controlador
- ▶ Permite programar diretamente o comportamento da rede pelo operador
 - ▶ Reduzindo tempo de mercado para soluções específicas e
 - ▶ Evitando ter que esperar que todos os fabricantes implementem a funcionalidade desejada



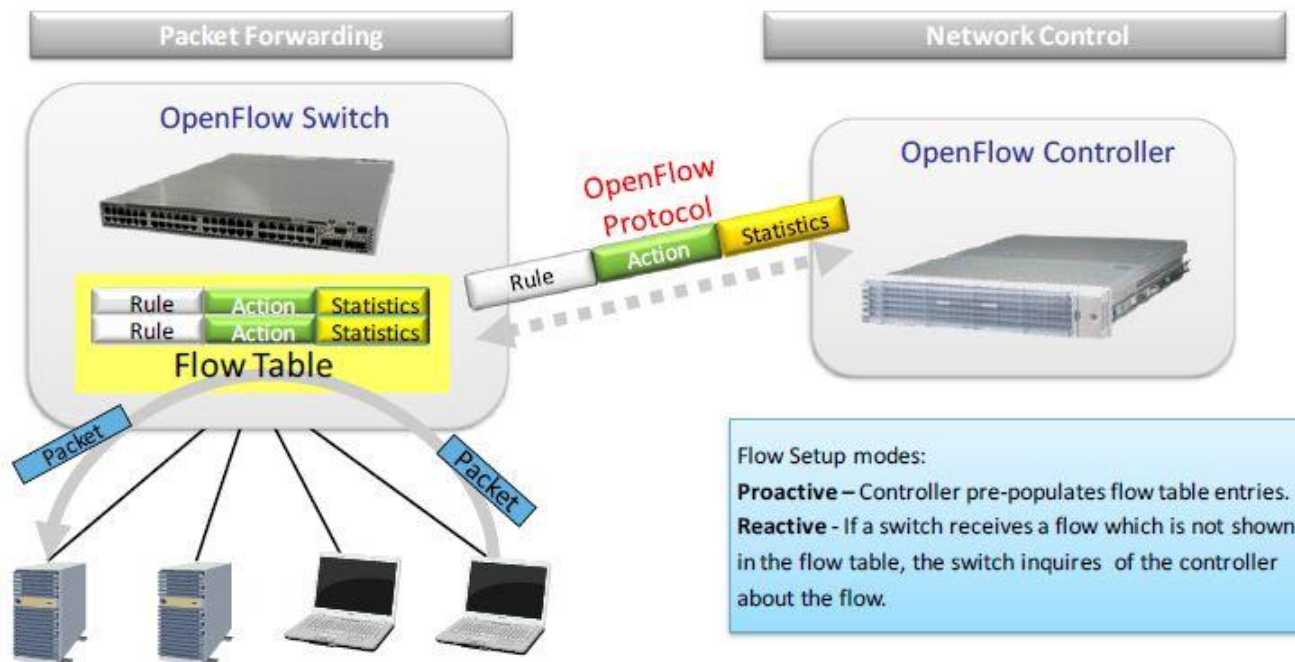
OpenFlow

- ▶ De volta as origens: comunicação entre os pontos cria um fluxo
 - ▶ OpenFlow permite controlar o roteamento do fluxo fim-a-fim
 - ▶ OpenFlow permite que implementem serviços dentro da rede

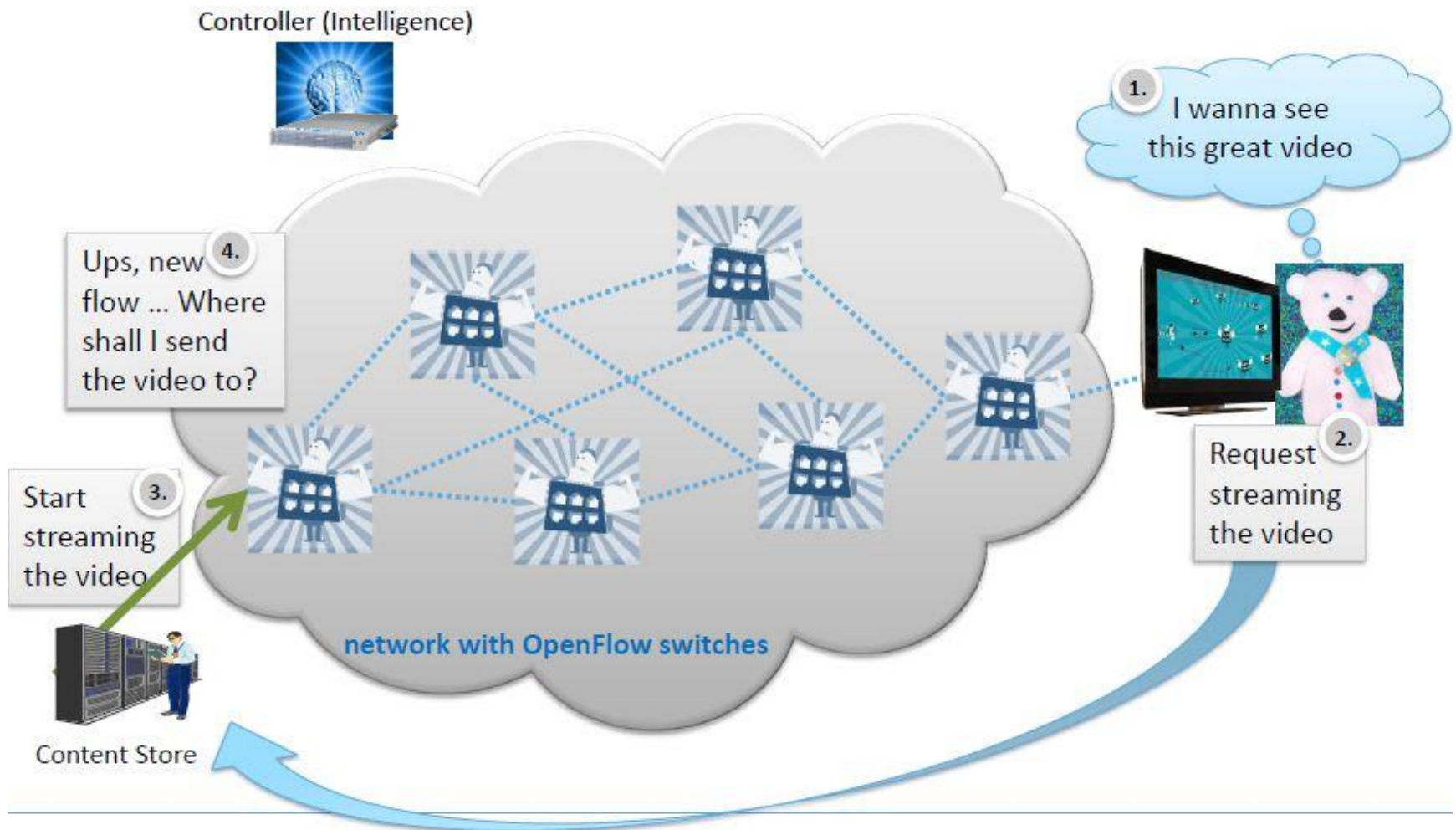


Como isso funciona?

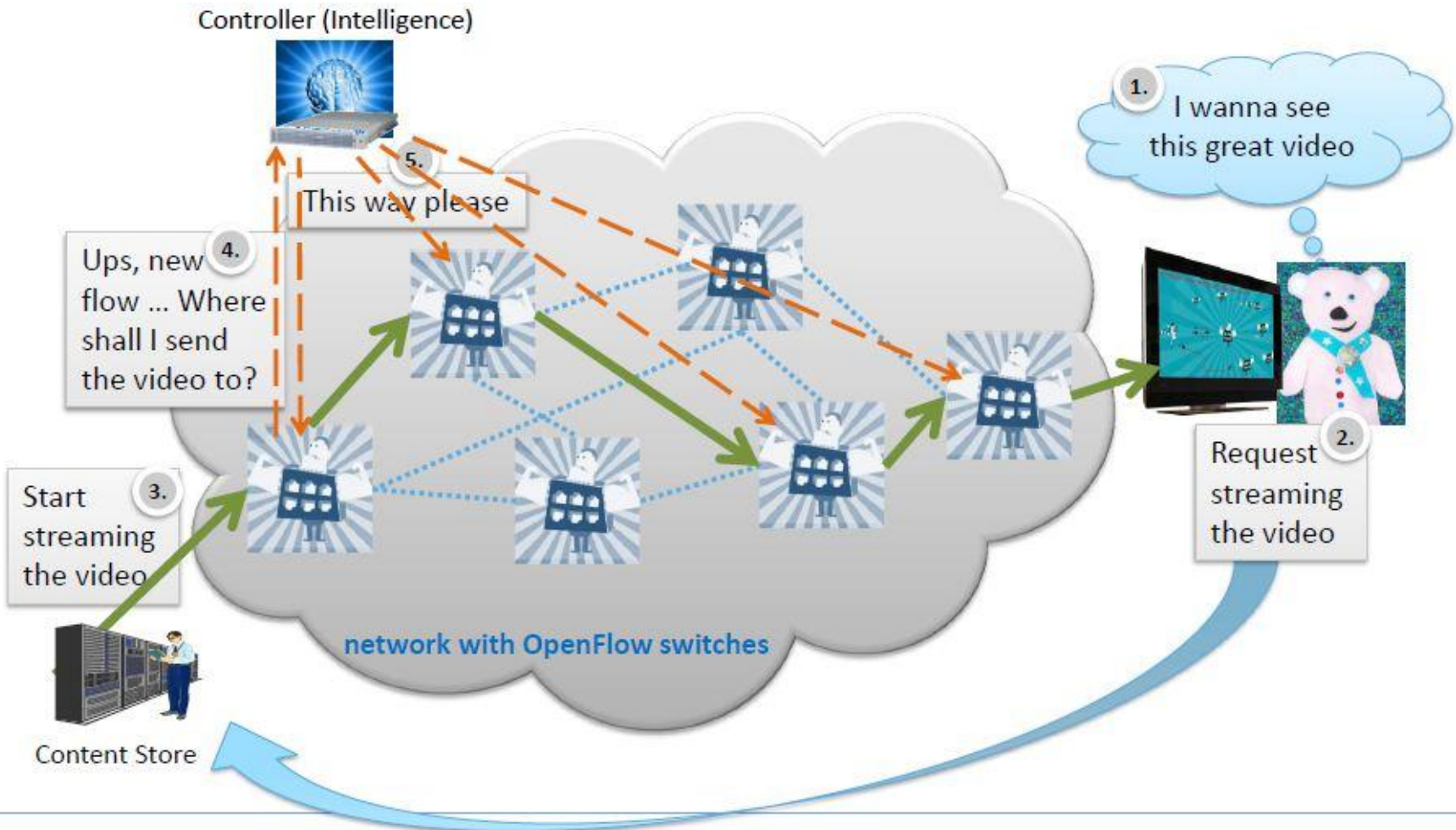
- ▶ Um switch OpenFlow somente redireciona *frames* se houver uma entrada na tabela de fluxo
- ▶ Caso não exista uma entrada, ele precisa perguntar ao controlador para obter uma entrada correspondente



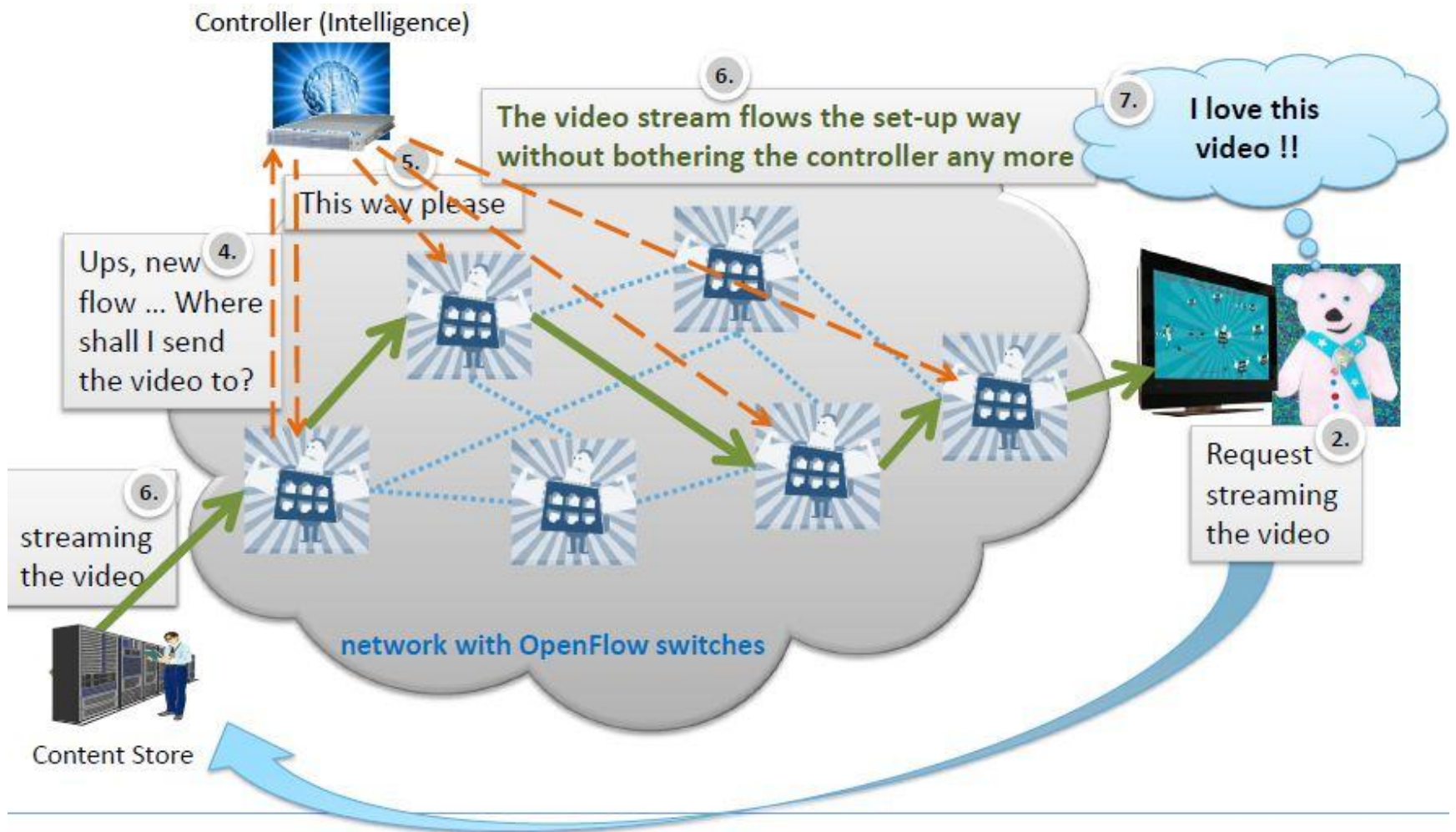
Exemplo (1)



Exemplo (2)

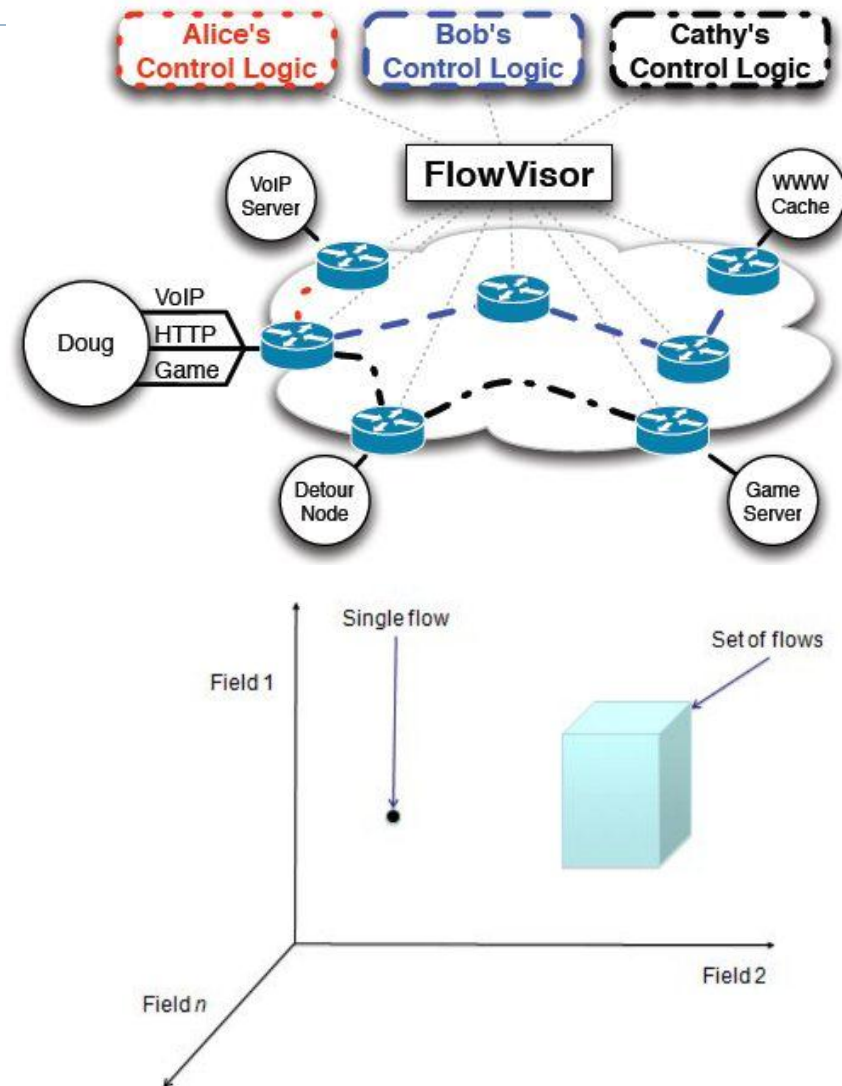


Exemplo (3)



FlowVisor

- ▶ FlowVisor é um tipo de **controlador OpenFlow** especial que age como um **proxy transparente** entre switches OpenFlow e múltiplos controladores OpenFlow
- ▶ **FlowVisor** cria “fatias” interessantes de recursos de rede e delega o controle de cada “fatia” para um controlador diferente
- ▶ FlowSpace: “**Fatias**” podem ser **definidas por quaisquer combinações de:** portas de switch (layer 1), endereço orig/dest ethernet ou tipo ou VLANs (layer 2), endereço IP orig/dst ou tipo (layer 3), e porta TCP/UDP orig/dest ou código/tipo ICMP (layer 4).
- ▶ **FlowVisor** garante **isolamento** entre cada “fatia”, de modo que um slice não possa afetar outro (ex. Produção)



OFELIA

- ▶ Provê uma **infraestrutura de experimentação única** baseada em OpenFlow
 - ▶ Permite aos pesquisadores não somente experimentar em uma rede de teste
 - ▶ Mas permite controlar a própria rede precisamente e dinamicamente
 - ▶ Permite a experimentação em redes multi-layer e multi-
tecnologia
- ▶ Programe a sua própria **rede em nuvem!**
- ▶ Conjunto de ilhas disponíveis para experimentos
- ▶ OFELIA Website:
<http://www.fp7-ofelia.eu/>



Eight Interconnected OpenFlow Islands

Ferramentas OFELIA

- ▶ Os **usuários controlam sua própria rede** e podem criar uma “Internet própria para eles”
- ▶ OFELIA **provê as ferramentas** para criar inovação na Internet do Futuro
- ▶ OFELIA está criando um **substrato Europeu** para esse novo tipo de maneira de pensar e lidar com redes, e rodar serviços nas mesmas

OFELIA Fluxo de Trabalho

1



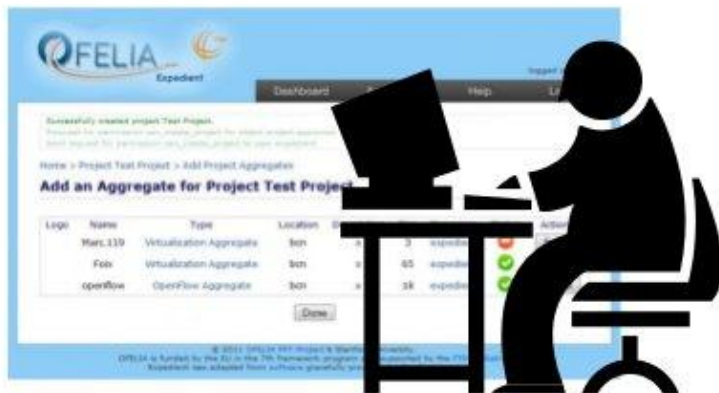
Have an idea!

2



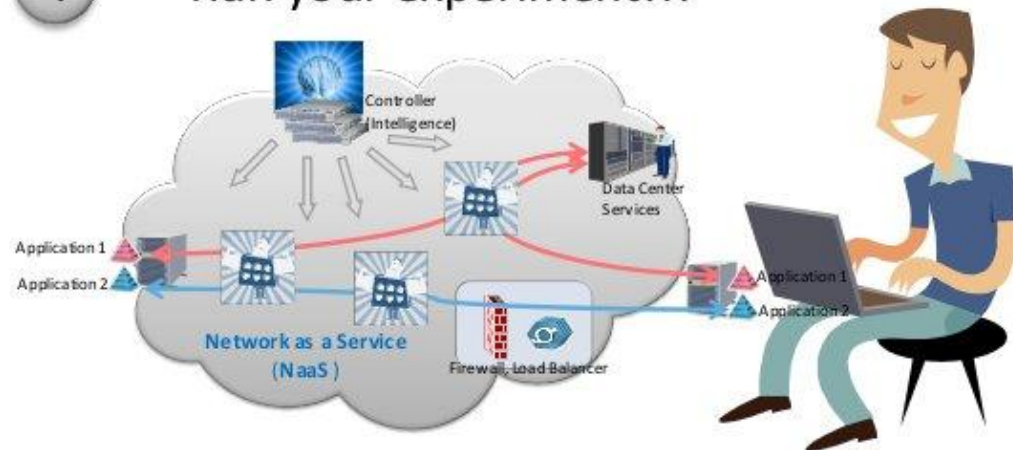
Log on to OFELIA!

3 Configure your network slice!

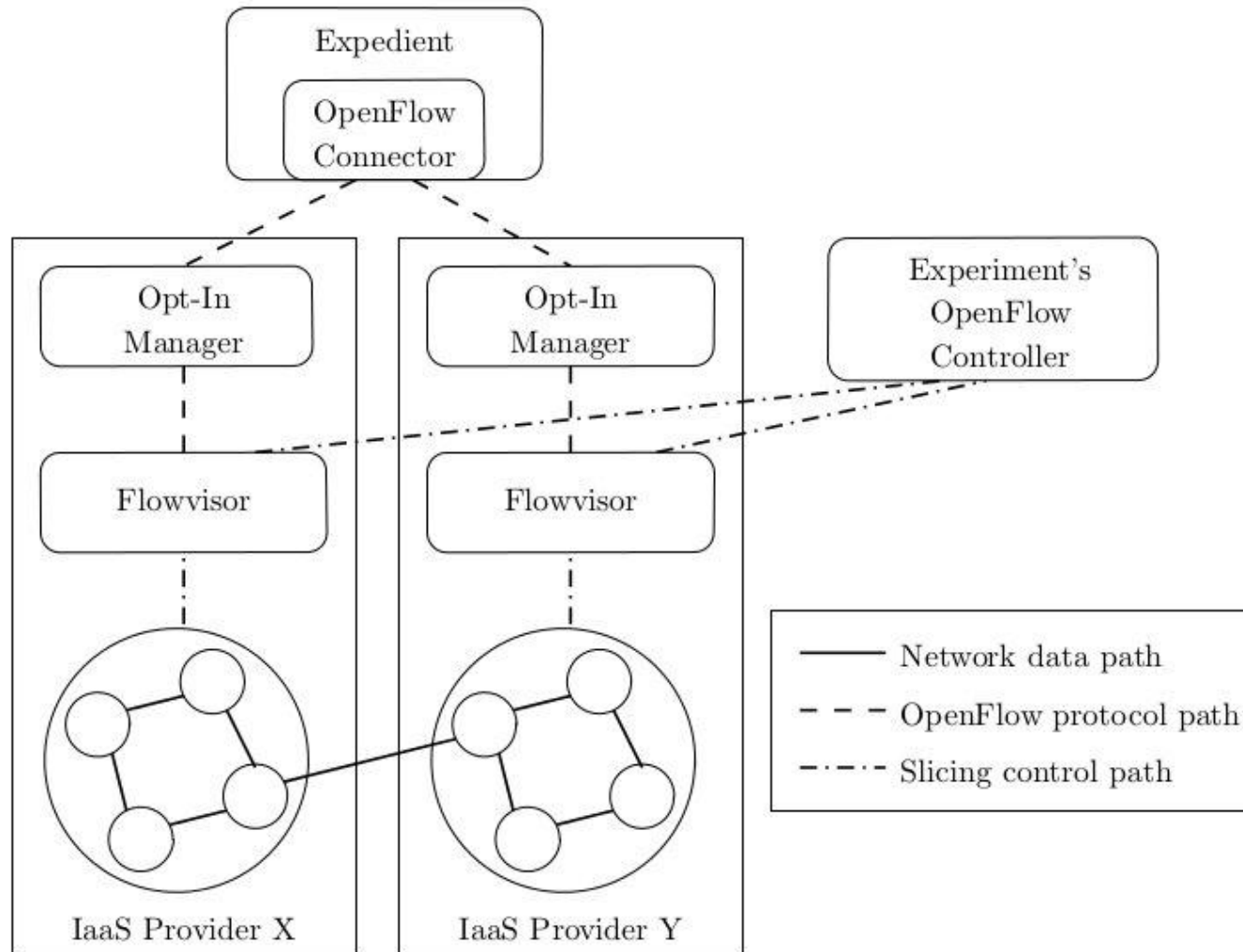


4

Run your experiment!!!



Baseado no Expedient/Opt-In



Arquitetura Interna do OFELIA

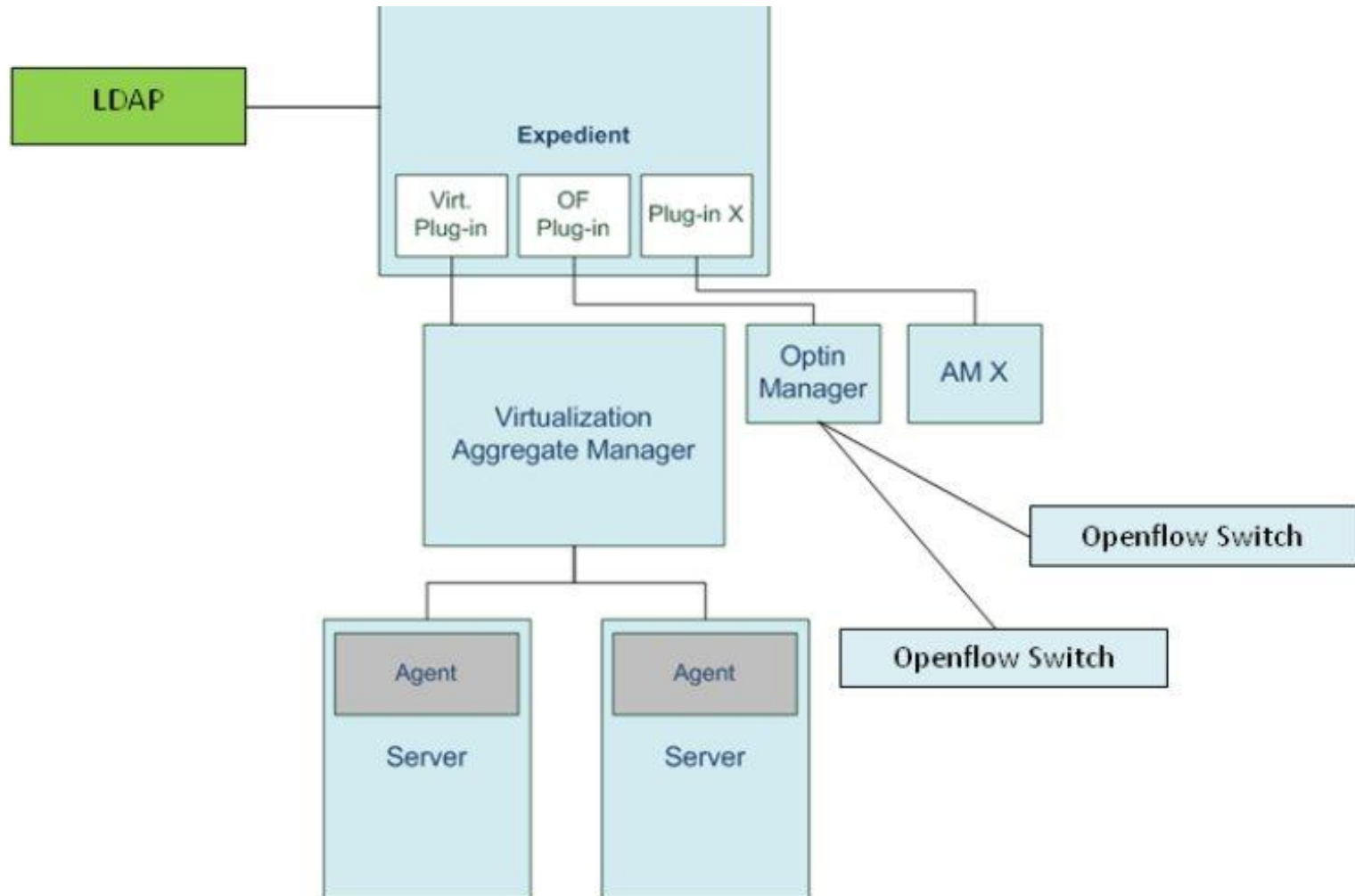


Figure 7: Expedient architecture with the additions.

Estrutura de uma ilha OFELIA

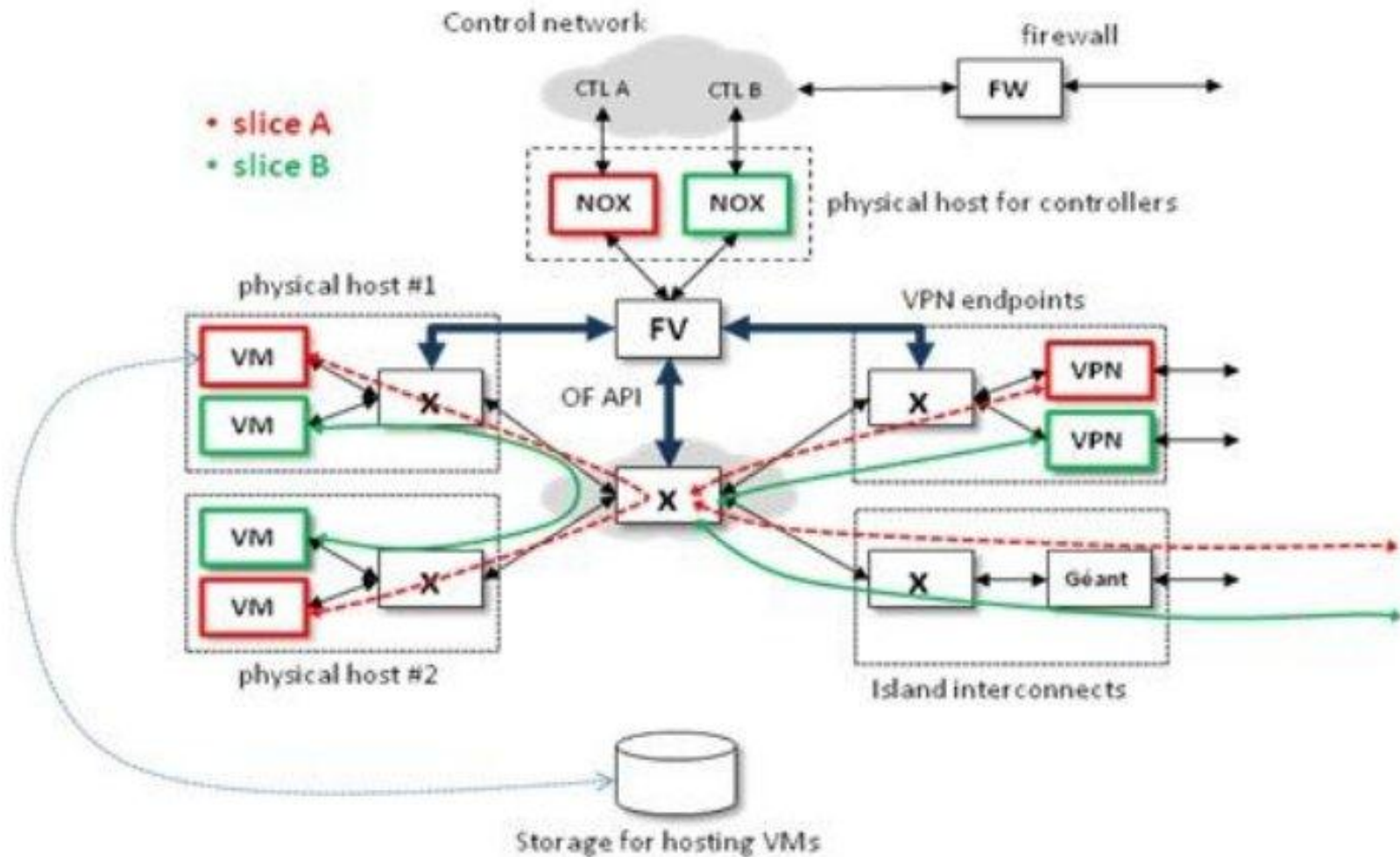


Figure 1: Isolated island basic scenario

OFELIA – ponto de vista usuário

- ▶ **Registrando uma conta:**

- ▶ https://alpha.fp7-ofelia.eu/doc/index.php/Getting_an_OFELIA_account

- ▶ **Configurando uma conexão OpenVPN:**

- ▶ https://alpha.fp7-ofelia.eu/doc/index.php/VPN_setup

- ▶ **Fazendo login em uma ilha OFELIA:**

- ▶ i.e.: <https://exp.i2cat.fp7-ofelia.eu/>

- ▶ **Criando um projeto:**

- ▶ Submetendo requisições de permissão

- ▶ Preenchendo campos de informação do projeto, adicionando membros, gerenciando “slices” do projeto



OFELIA – ponto de vista usuário

- ▶ **Adicionando Agregados;**
 - ▶ Opt-in /VT_Manager (pelo menos 1 de cada)
- ▶ **Criando um Slice:**
 - ▶ Preenchendo um nome ao slice, descrição, TTL e adicionando Agregados;
- ▶ **Gerenciando os Recursos:**
 - ▶ Criando VMs (nome das VM, selecionando image, memória);
- ▶ **Configurando VMs:**
 - ▶ É preciso acessar via SSH, e instalar e configurar ferramentas mais tarde;



OFELIA – ponto de vista usuário

- ▶ **Reservando (Booking) os Recursos:**
 - ▶ Selecionando portas e switches, configurando flowspace;
- ▶ **Configurando o Controlador daquele Slice:**
 - ▶ Ele pode ser dentro de uma VM ou remotamente através de VPN;
 - ▶ Preencher um Endereço IP, porta, e senha (para uso com controlador autenticado);
- ▶ **Aguardar o Aceite do Administrador**
- ▶ **Iniciar o Slice.**



Demonstração por Vídeo

- ▶ <http://www.dc.ufscar.br/~marcondes/ofelia.ogv>

OFELIA - Básico de Instalação

▶ Dependências

- ▶ Debian Squeeze (recomendado)
- ▶ Python 2.x
- ▶ MySQL, Flowvisor (CF)
- ▶ XEN kernel and userland (Agent)

- ▶ Tarball a partir do site Ofelia's Codebasin, unzip to /opt/ofelia

<http://codebasin.net/redmine/projects/ocf>



OFELIA - Básico de Instalação

▶ Arcabouço de Controle

▶ Configurando a rede Servidor

```
auto eth0
iface eth0 inet static
address 192.168.254.193
netmask 255.255.255.0
```

```
auto eth1.999
iface eth1 inet static
address 172.16.1.101
netmask 255.255.255.0
vlan_raw_device eth1
```

▶ Instalação do Flowvisor - Máquina Servidor

```
vim /etc/apt/sources.list (repositório non-free habilitado)
apt-get update
apt-get install sun-java6-jre sun-java6-jre
apt-get install sun-java6-plugin ant
update-java-alternatives -s java-6-sun
```

```
apt-get install build-essential autoconf automake libtool git
git clone git://gitoris.stanford.edu/flowvisor.git
cd flowvisor
make
adduser flowvisor
make install -prefix=/usr
update-rc.d flowvisor defaults
/etc/init.d/flowvisor start
```



OFELIA - Básico de Instalação

- ▶ Arcabouço de Controle
 - ▶ Configurando MySQL

```
apt-get install mysql-server
mysql -p
create user 'fibre' identified by 'senha';
create database expedient;
create database optin;
create database vt_am;
grant all on expedient.* to 'fibre' identified by 'senha' ;
grant all on optin.* to 'fibre' identified by 'senha' ;
grant all on vt_am.* to 'fibre' identified by 'senha' ;
\q
```

- ▶ Instalando o arcabouço de controle
 - Durante a instalação, requisita-se a edição de arquivos de configuração do OFELIA CF. Os parâmetros a serem editados definem o nome de usuário, senha e e-mail de um usuário administrador inicial, dados para conexão com o banco de dados, assim como o IP e porta para a interface do componente.

```
cd /opt/ofelia/expedient/bin
ofver install -f
cd /opt/ofelia/optin_manager/bin
ofver install -f
cd /opt/ofelia/vt_manager/bin
ofver install -f
```



OFELIA - Básico de Instalação

▶ Agente (OXA – OFELIA Xen Agent)

- ▶ Configurando a rede Agente (parecida com a do servidor)
- ▶ Carregando os modulos adicionais no boot (loop, 8021q)

▶ Instalando Xen

```
# Distribuição x86
apt-get install xen-linux-system-2.6-xen-686 vlan python-libvirt libvirt-dev xen-tools
# Distribuição x86_64
apt-get install xen-linux-system-2.6-xen-amd64 vlan python-libvirt libvirt-dev xen-tools
```

▶ Editando as configurações do XEN e alterando a BRIDGE

```
(xend-http-server yes)
(xend-port 8000)
# 'network-multi-bridge-vlan' manualmente
(network-script 'network-multi-bridge-vlan')
```

```
export
SITE=http://www2.comp.ufscar.br/~ricardofg/OFELIA
wget $SITE/network-bridge-vlan -o \
/etc/xen/scripts/network-bridge-vlan
wget $SITE/network-multi-bridge-vlan -o \
/etc/xen/scripts/network-multi-bridge-vlan
chmod +x /etc/xen/scripts/*
wget $SITE/xend -o /etc/init.d/xend
```



OFELIA - Básico de Instalação

▶ Preparando a instalação do software Agente

```
mkdir -p /opt/OFELIA/oxa  
ln -s /opt/OFELIA /opt/OFELIA/oxa/repository
```

▶ Criando uma VM template e edição do script ofver

▶ Instalando o software

```
cd  
/opt/OFELIA/oxa/repository/vt_manager/src/python/agent/tools  
./ofver install
```

```
cd /etc/xen-tools/role.d  
wget $SITE  
wget $SITE/xen/OFELIA-helper.sh  
chmod +x OFELIA*  
xen-create-image -passwd -role=OFELIA -install-method=debootstrap -  
dist=squeeze -ip=172.16.1.99  
-netmask=255.255.255.0 -hostname=fibre-default  
mkdir /home/vm-images  
mkdir /mnt/xen  
mount -o loop /home/xen/domains/algum_hostname/disk.img /mnt/xen  
cd /mnt/xen  
tar pcgz /home/vm-images/default.tar.gz *  
umount /mnt/xen  
cd /home/vm-images/  
md5sum default.tar.gz » default.hash
```

Configurando OFELIA como Administrador

- ▶ Criar faixas de IP para máquinas virtuais e/ou faixas de MAC ethernet virtuais
- ▶ Adicionar no Expedient a comunicação com OXA
- ▶ Especificar outros detalhes (Management Bridges e Data Bridges)
- ▶ Configurar o Flowvisor
- ▶ Configurar a Clearinghouse (ex. LDAP)
- ▶ Aprovar uma requisição de FlowSpace
- ▶ Adicionar e editar regras de FlowSpace



Cesar Marcondes

marcondes@dc.ufscar.br

Os autores gostariam de agradecer às equipes FIBRE-BR da UFSCar, UFPA, UNIFACS, UFG, USP e demais membros do FIBRE-BR. Também agradecemos a Marcial Fernandez (UECE), Jorge Barros (CPqD), Mateus Cerezini (UFES).

<http://www.fibre-ict.eu/>



FUTURE INTERNET TESTBEDS
EXPERIMENTATION BETWEEN
BRAZIL AND EUROPE