



```
#endif
// Error include 'skk.h' before including
#include "resource.h" // make resource
// See DMotion.cpp for the implementation of
class CDMotionApp : public CWinApp
{
public:
    CDMotionApp();
    // Overrides
    // ClassWizard generated override
    virtual BOOL InitInstance();
};
```



BGP Connectivity for Virtual Networks

Vytautas Valancius, Yogesh Mundada,
and Nick Feamster

**Georgia
Tech**



College of
Computing

School of Computer Science

Emerging Network Infrastructure

- Network Virtualization



- Cloud Computing





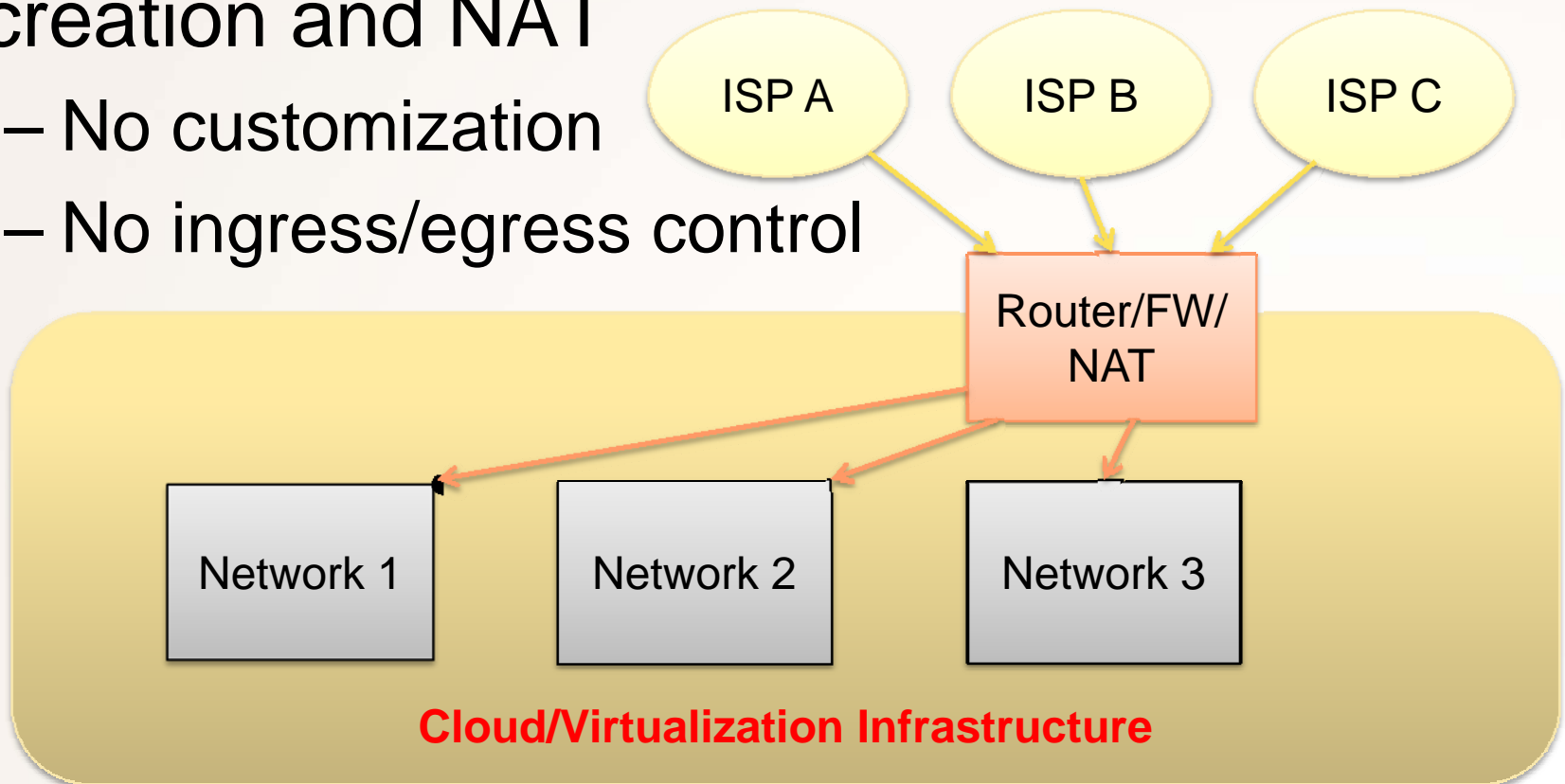
Cloud Services and Virtual Networks are at the Rudimentary Stage

- Virtual topologies inside the cloud
 - Need for fast resource allocation algorithms
 - **Need for flexible topology mapping and resource description languages**
- External connectivity
 - **Need for a greater user control**
 - **Need for flexible interconnects**



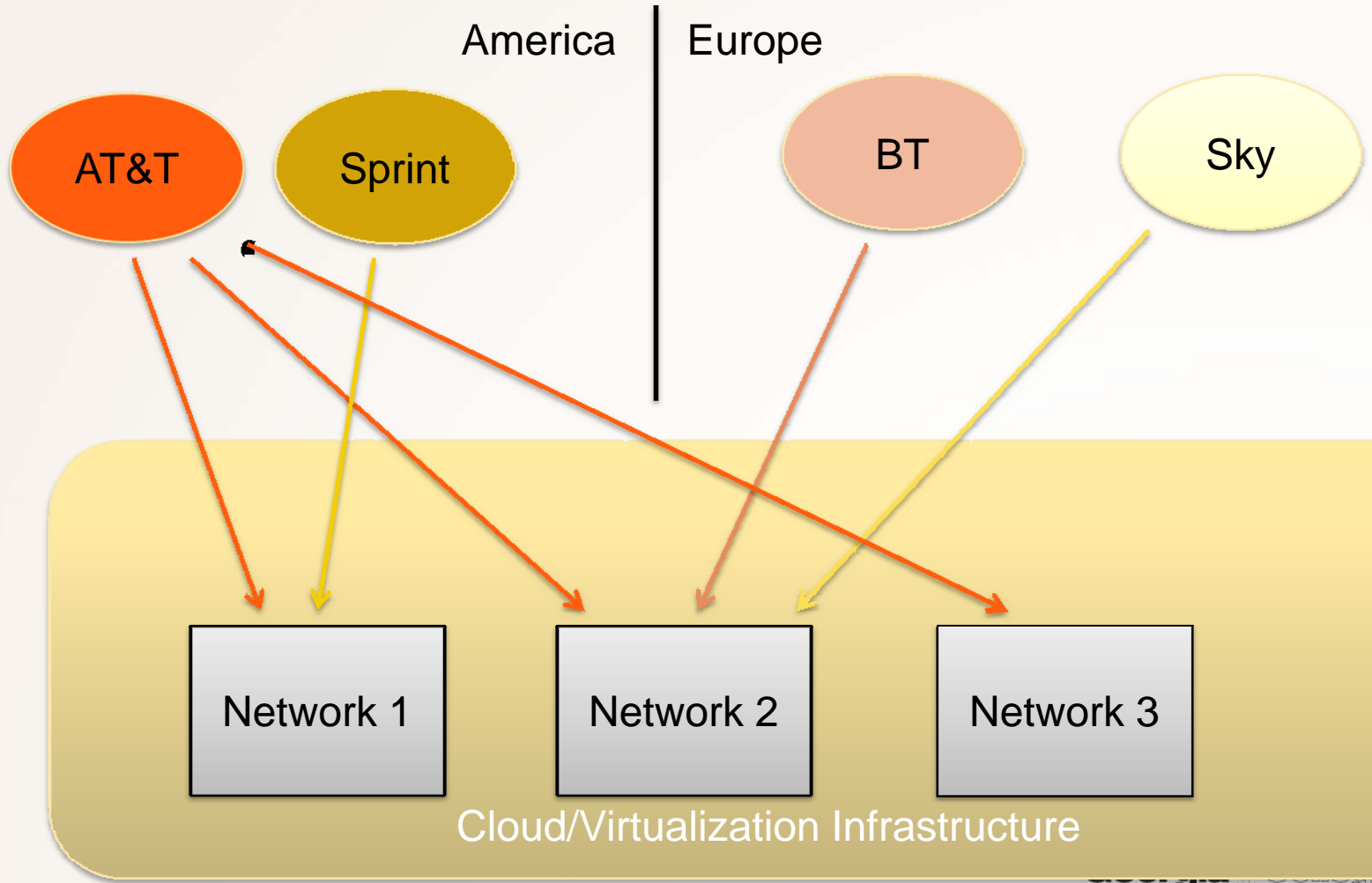
Virtual Networks Need Connectivity

- Strawman solution: manual topology creation and NAT
 - No customization
 - No ingress/egress control





Instead: Appearance of Direct Connectivity

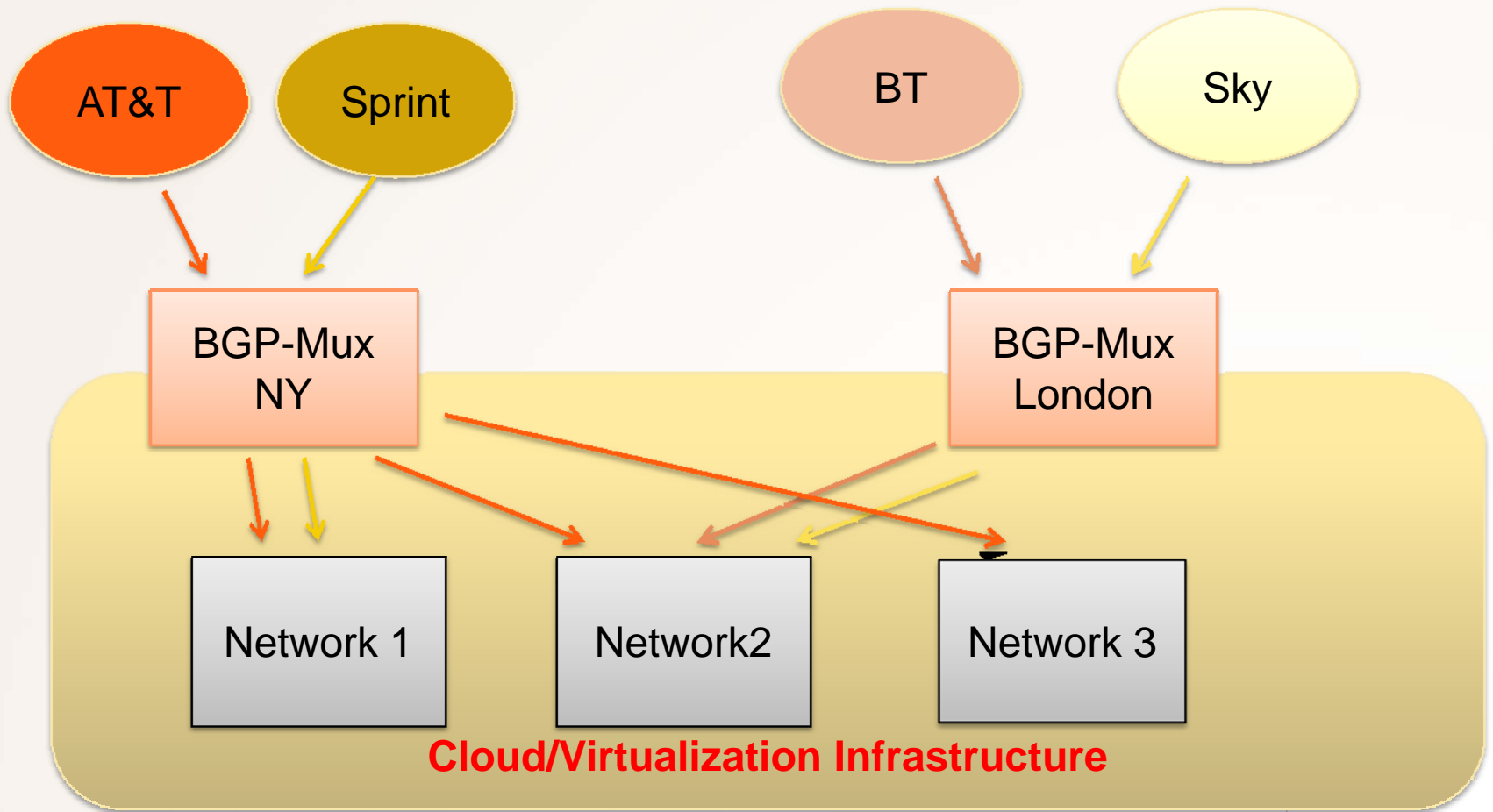




Challenges for Direct Connectivity

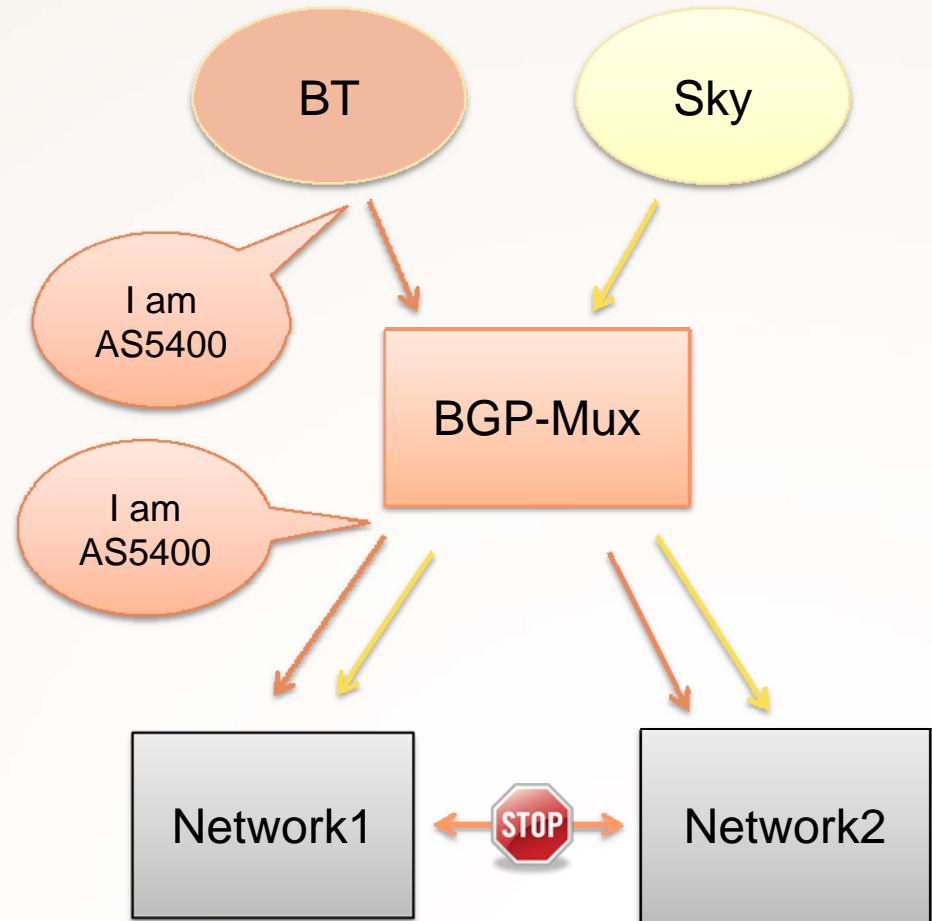
- Lack of stability
 - Virtual networks come and go
 - ISPs are unwilling to keep configuring BGP sessions
 - Virtual network users need to negotiate with multiple ISPs
- Lack of control
 - Cloud provider need more control for accounting

Solution: "BGP Mux"

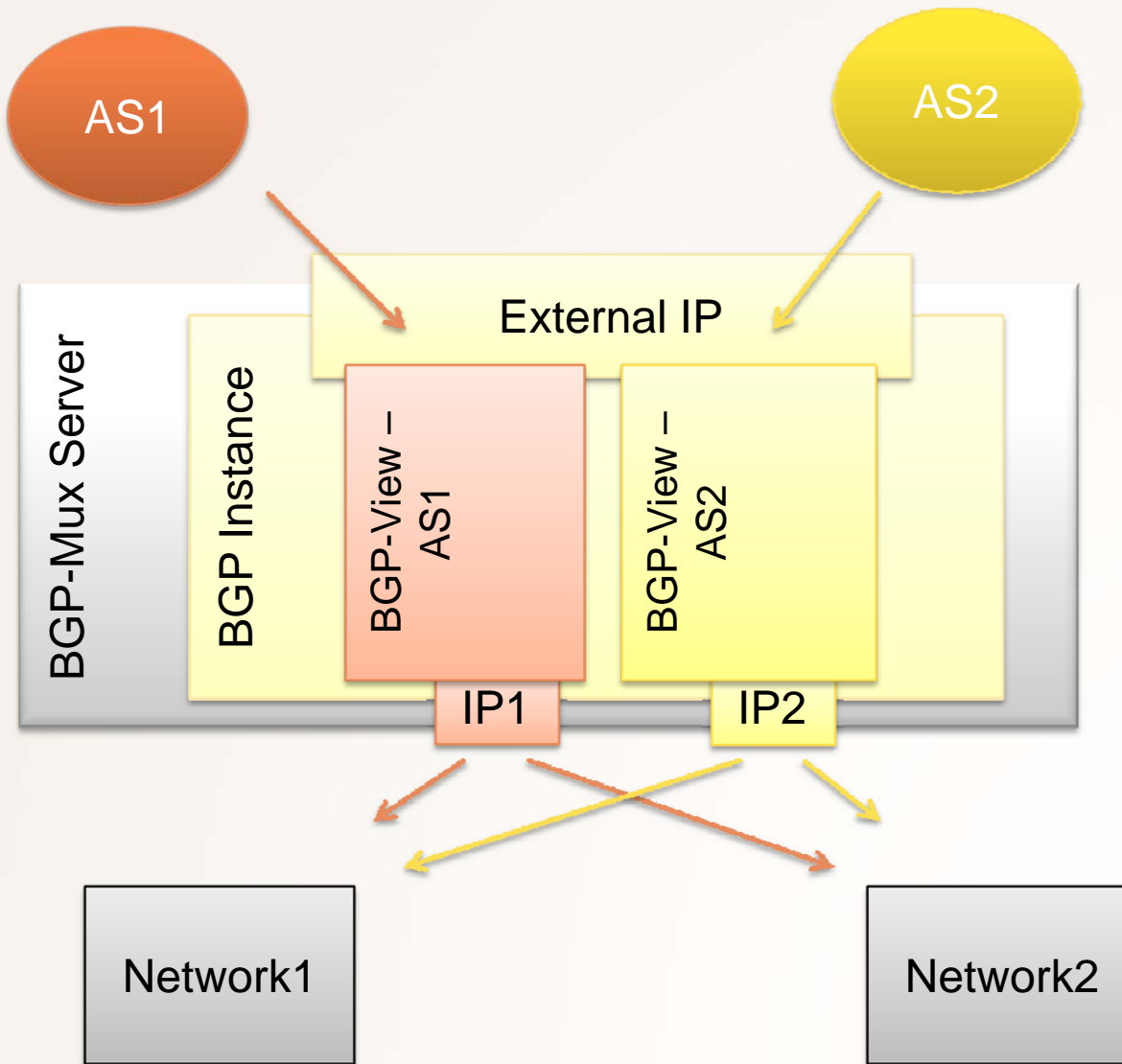


Design Requirements

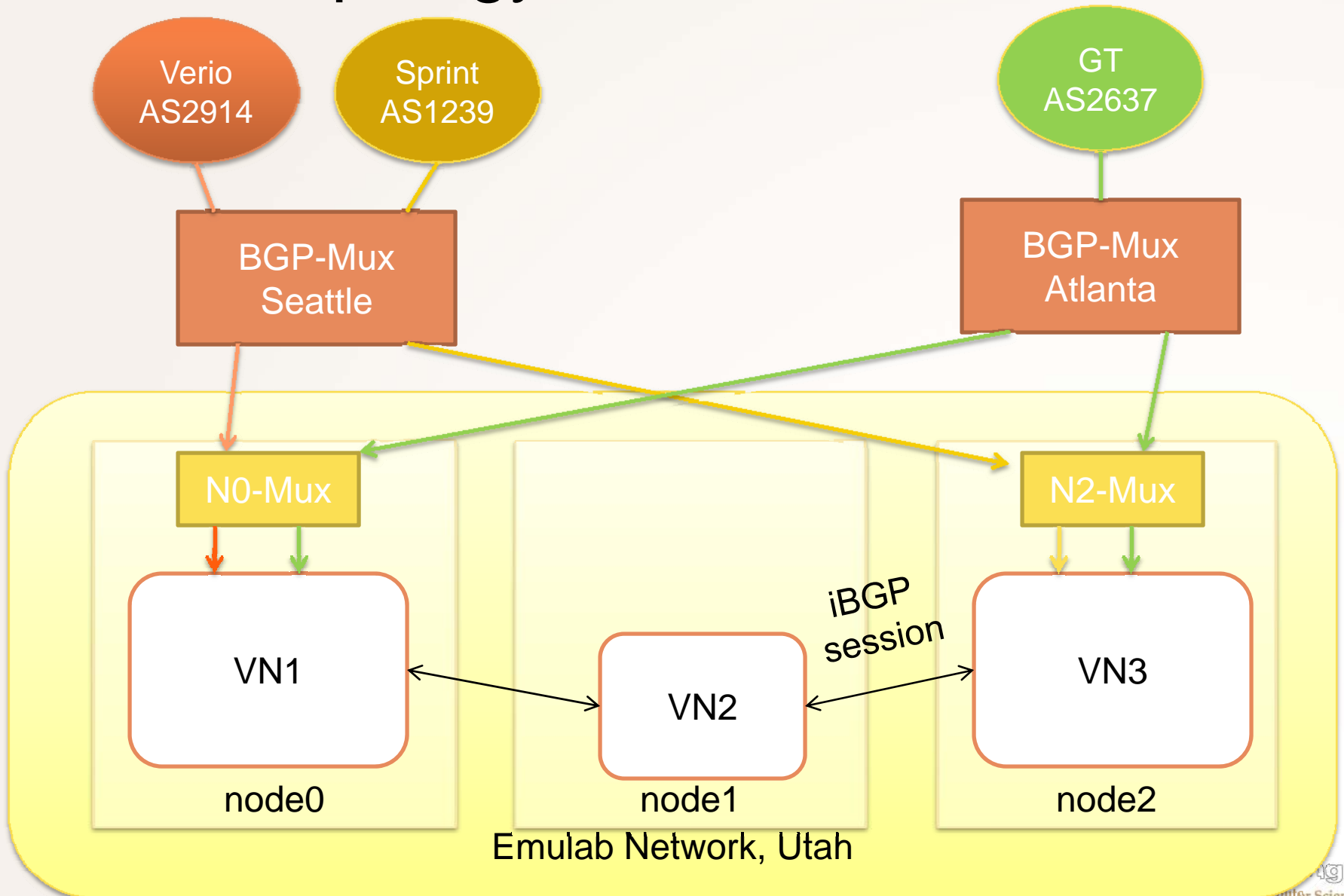
- Session transparency
 - User thinks it connects to an ISP
- Session stability
 - Fluctuating user sessions are not observed by ISP
- Update transparency
 - Updates are passed unmodified.
 - No best route selection
- Isolation
 - No route leaking between the ISPs
- Scalability



Implementation: BGP Views



Demo: Topology



Demo

