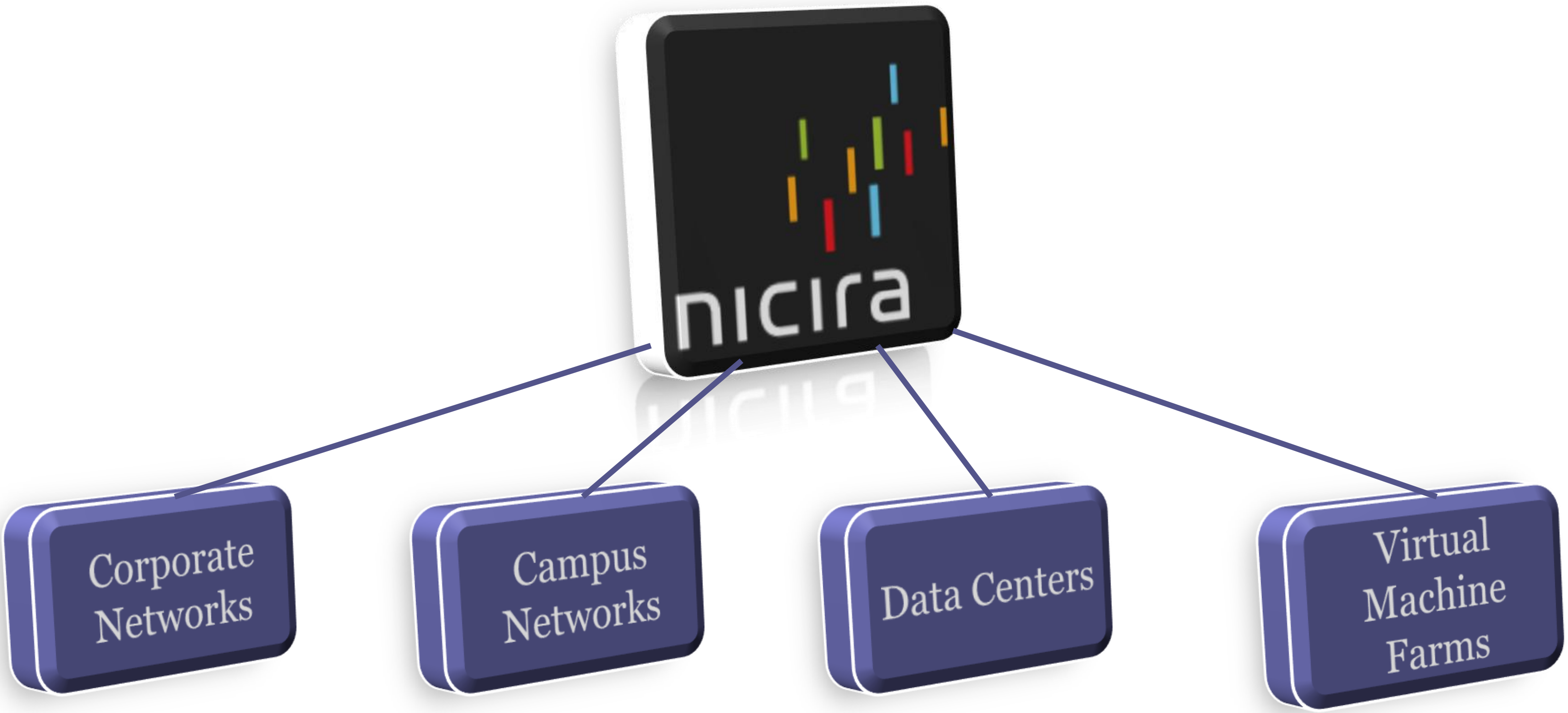
nox at Home

Andrew Arminio
Christopher Austin
James “Murphy” McCauley

Who is our Client?

Nicira develops infrastructure technologies for large networks



Infrastructure technologies?

What does that *mean*?

- Load balancing
- Client traffic isolation for multi-tenant data centers
- Virtual machine migration
- Facilitating cloud computing

What are OpenFlow and NOX?

- Two technologies that Nicira uses
- Work together
- A platform for writing network control software
  - Using a centralized programming model
  - Using a C++ or Python API

Nicira’s task for our team

What about NOX and OpenFlow on small networks?

- **Home Networks**
- **Small Office Networks**

Basic routing capability

NOX and OpenFlow

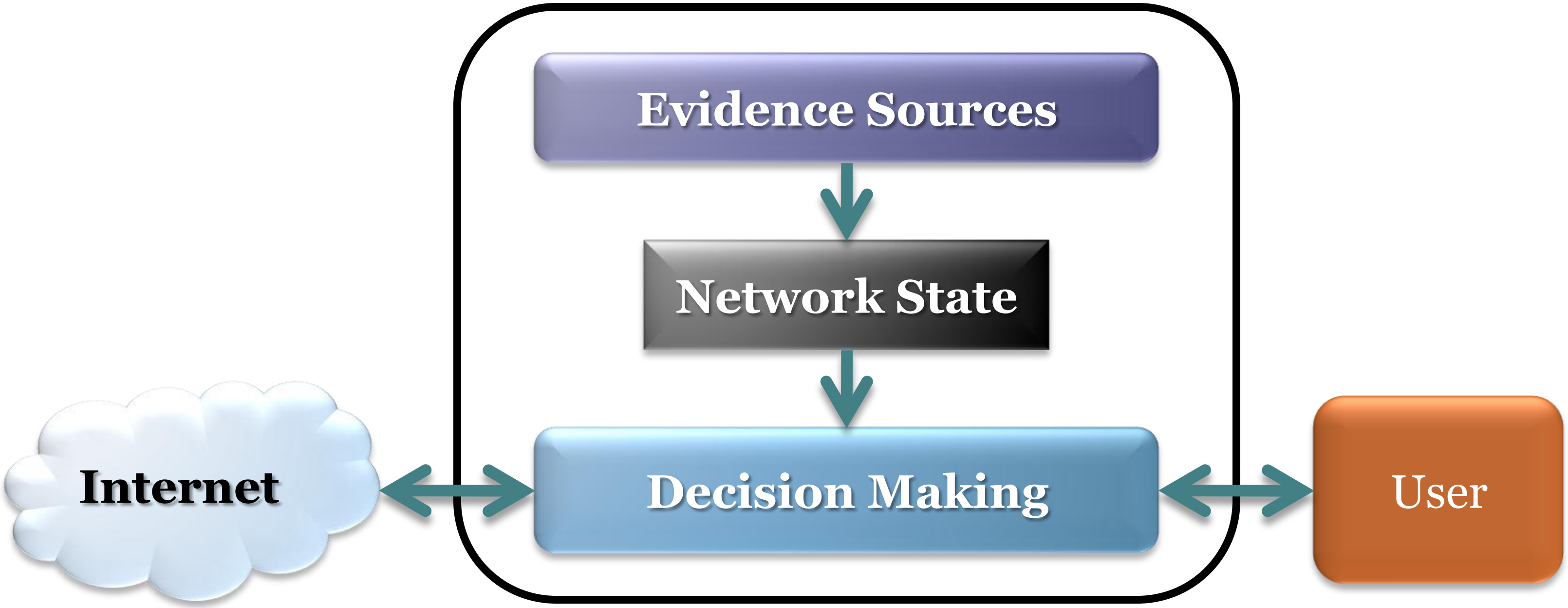
Possibilities…

- Bandwidth control
- Isolation of guest users
- Better parental control
- Centralized network diagnostics
- Password protected file sharing that works
- And more …

Making it happen

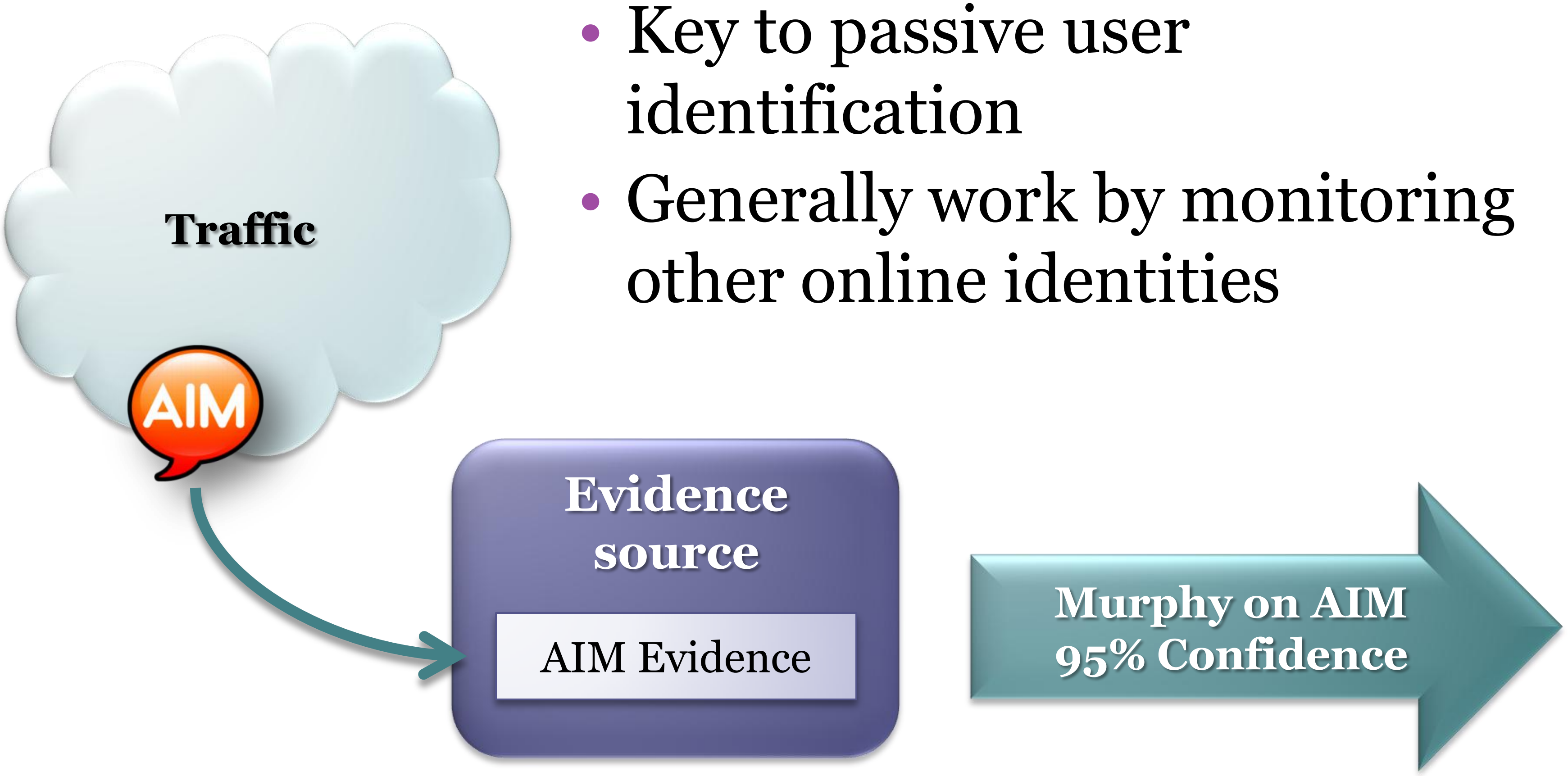
- Python Components
- A central concept: Passive User Identification
  - Much of what people do online is already tied to an identity
  - We can leverage these!
- Two major architectures
  - Controller Architecture
  - User Interface Architecture

Controller Architecture

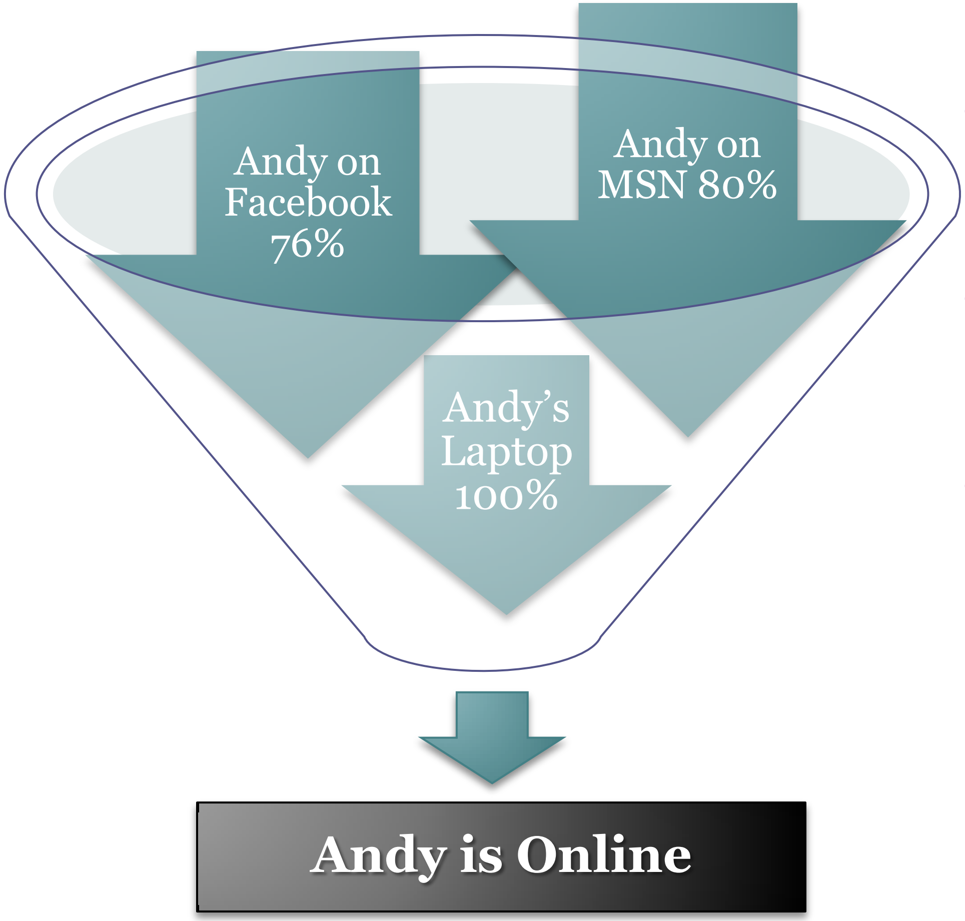


Evidence Sources

- Key to passive user identification
- Generally work by monitoring other online identities

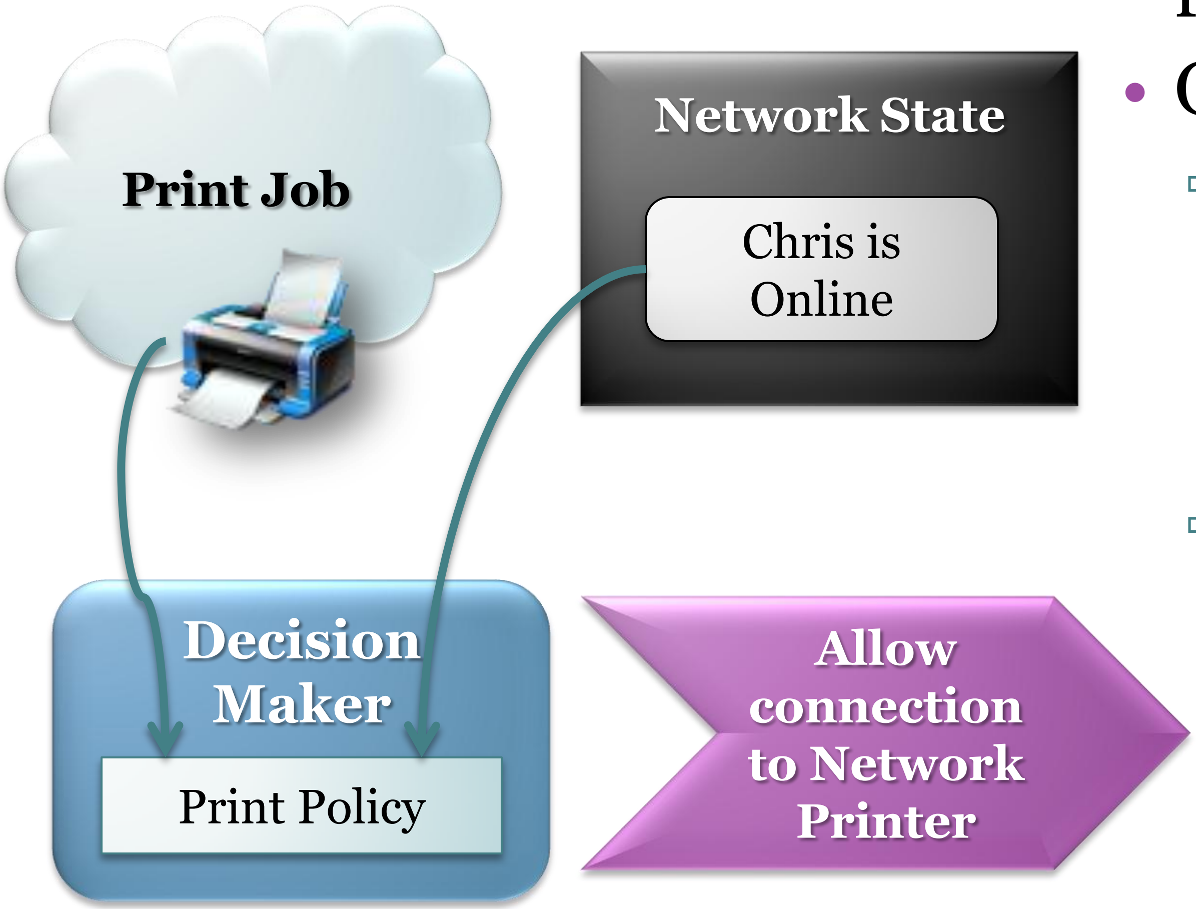


Network View (Judge & State)



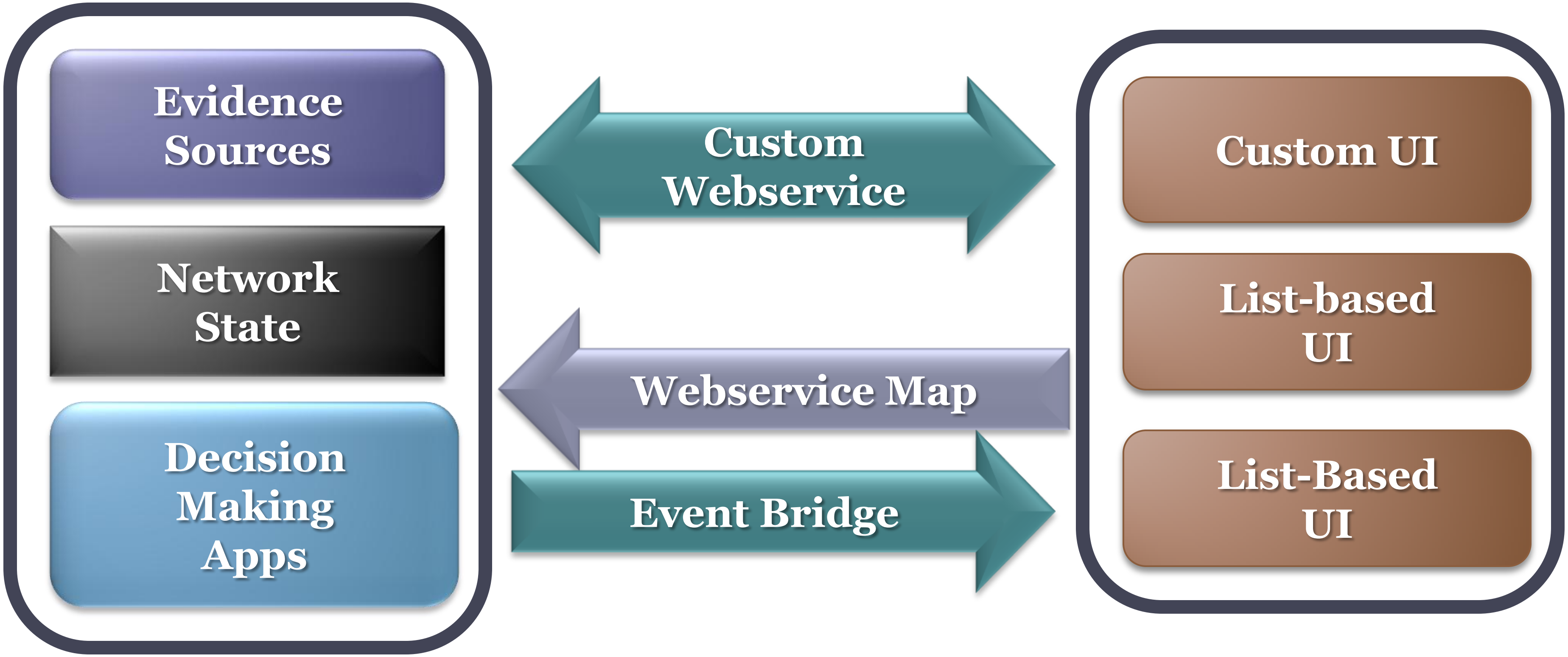
- Gathers evidence from Evidence Sources
- Makes determinations about Network State
- Provides state information for decision making

Decision Making



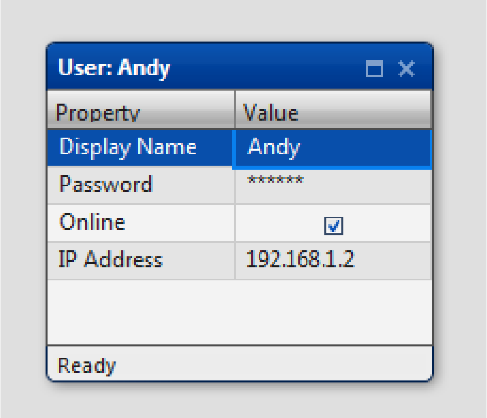
- Can take many forms
- One common class:
    - Associates traffic with a user or machine using Network View
    - Controls traffic to implement a policy

User Interface Architecture



List Based UI

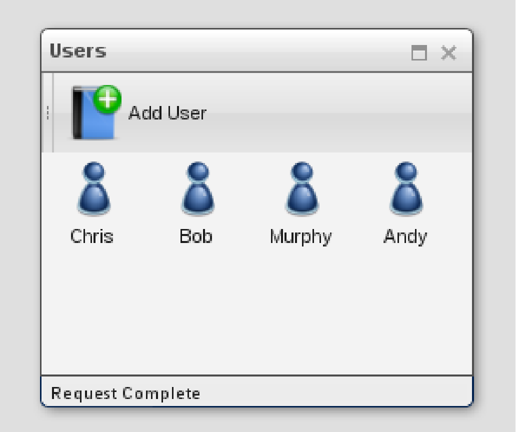
- Many user interfaces made of lists
  - Users
  - Configuration options
- Creates a web UI using Python lists and dictionaries



User: Andy

Property	Value
Display Name	Andy
Password	******
Online	☑
IP Address	192.168.1.2

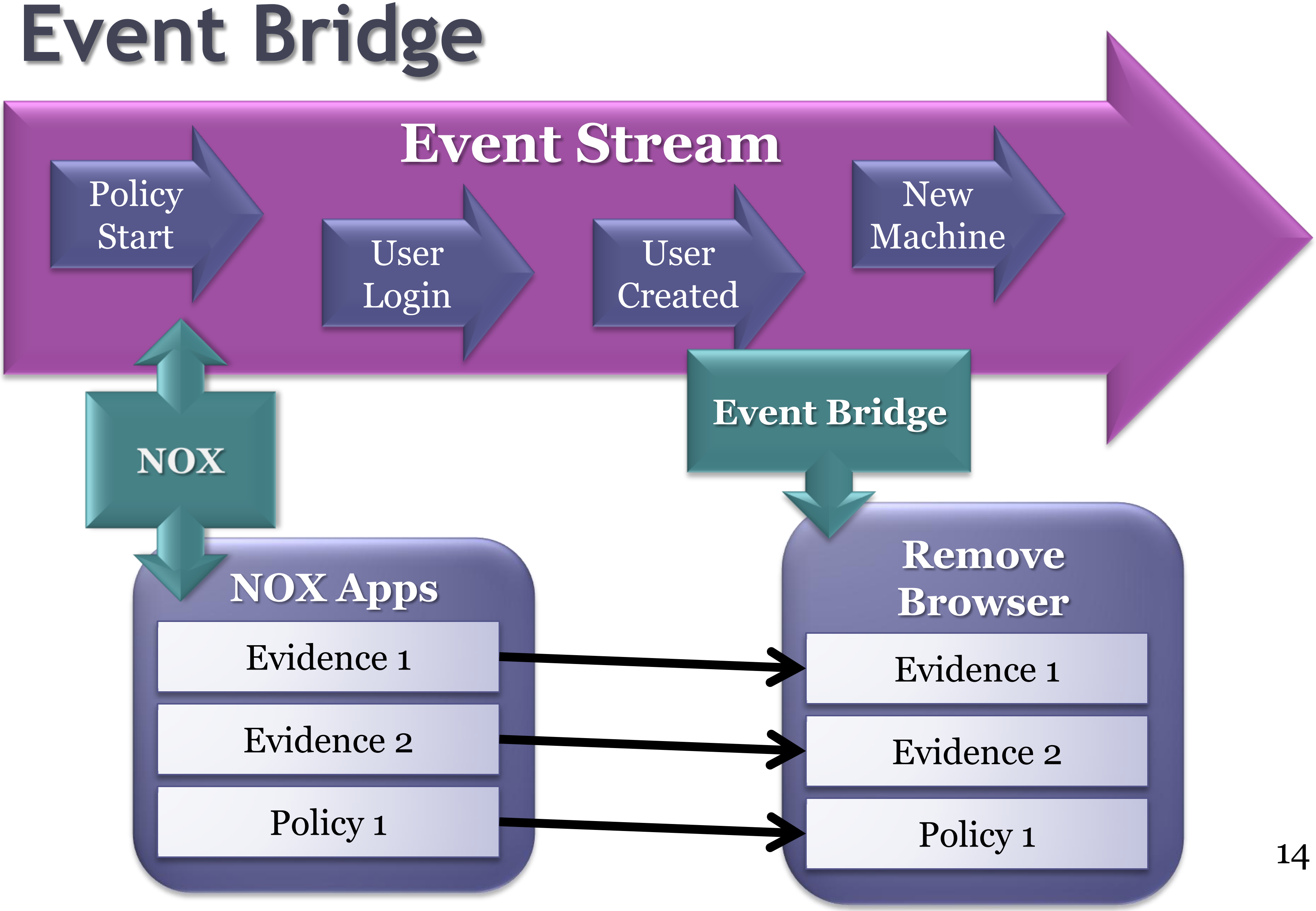
Ready



Users

Request Complete

Event Bridge



Event Stream

Policy Start

User Login

User Created

New Machine

NOX

Event Bridge

NOX Apps

Evidence 1

Evidence 2

Policy 1

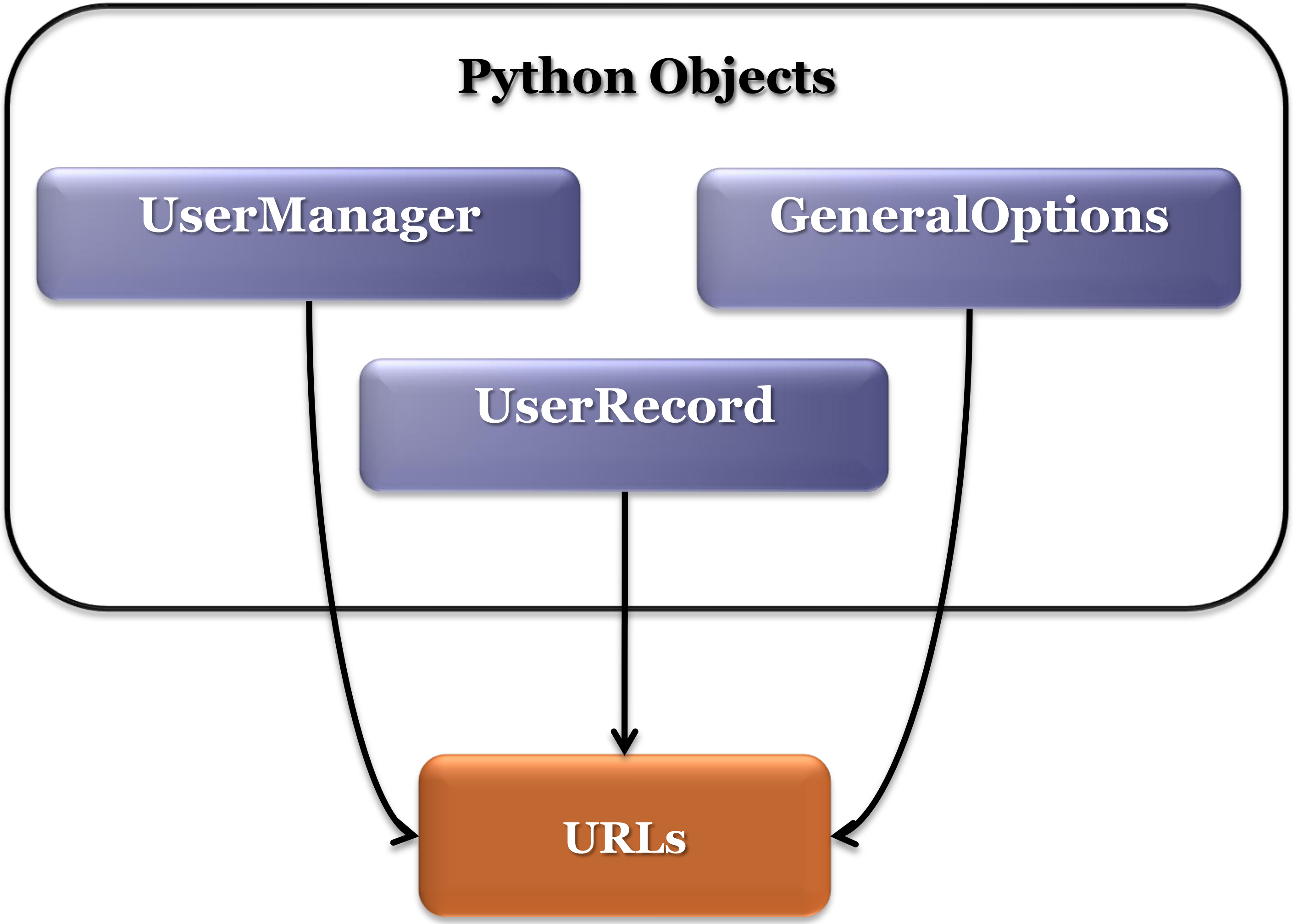
Remove Browser

Evidence 1

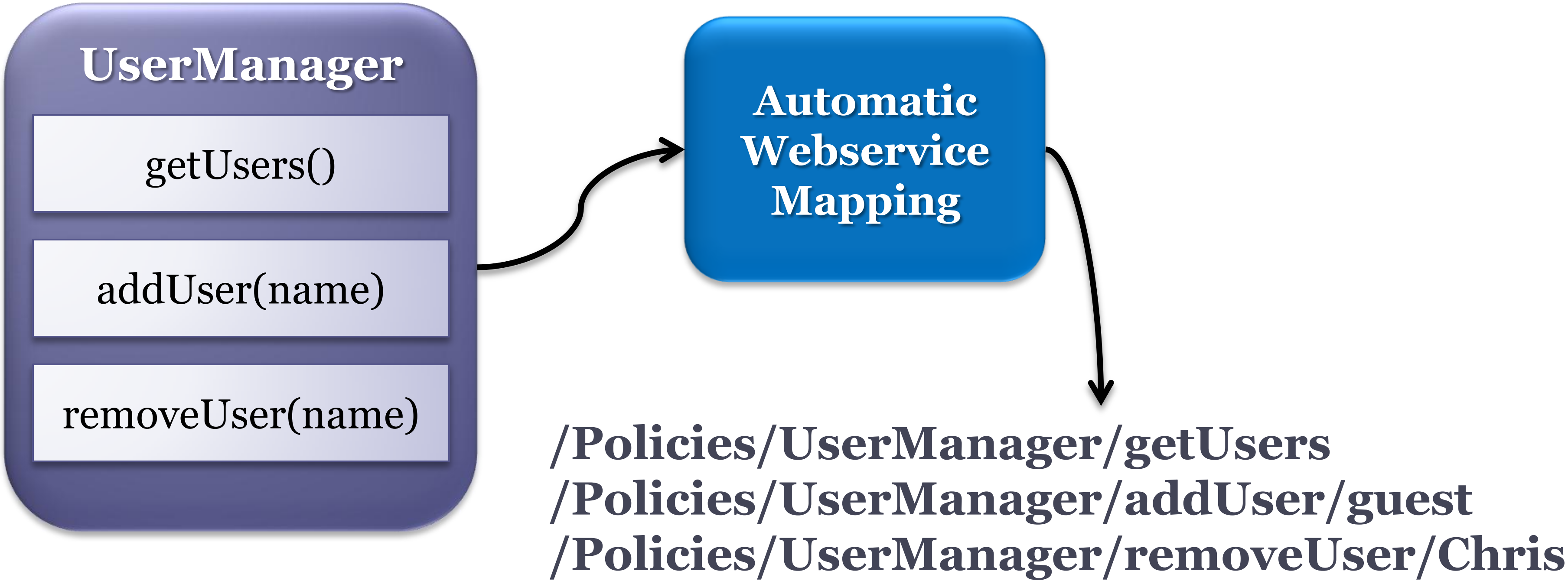
Evidence 2

Policy 1

Automatic Webservice Mapping



Automatic Webservice Mapping



UserManager

getUsers()

addUser(name)

removeUser(name)

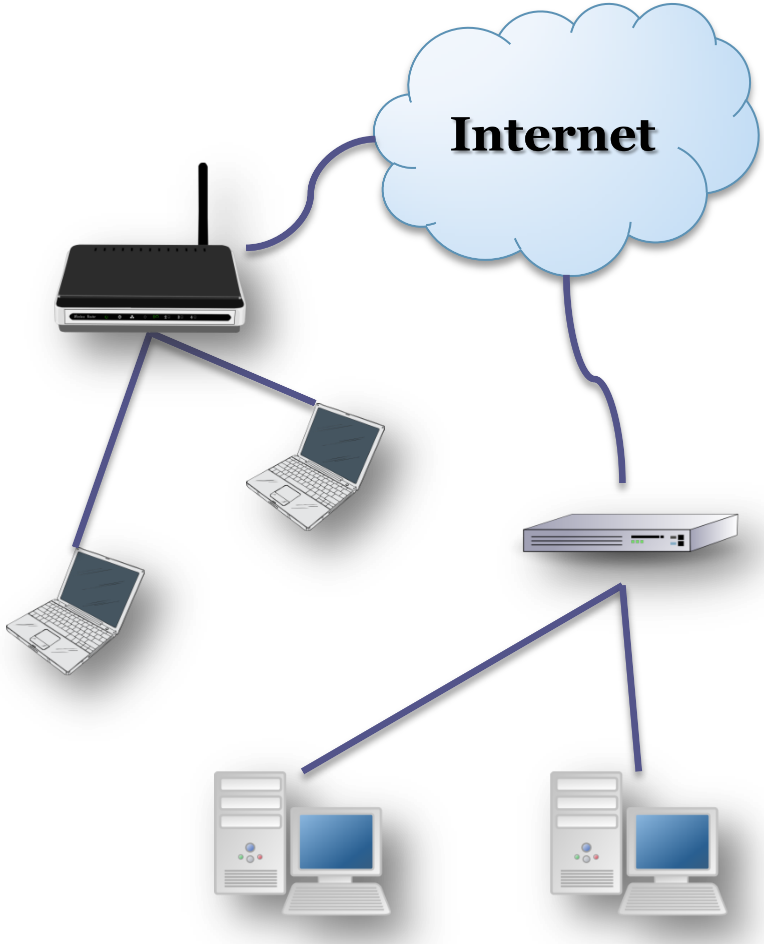
Automatic Webservice Mapping

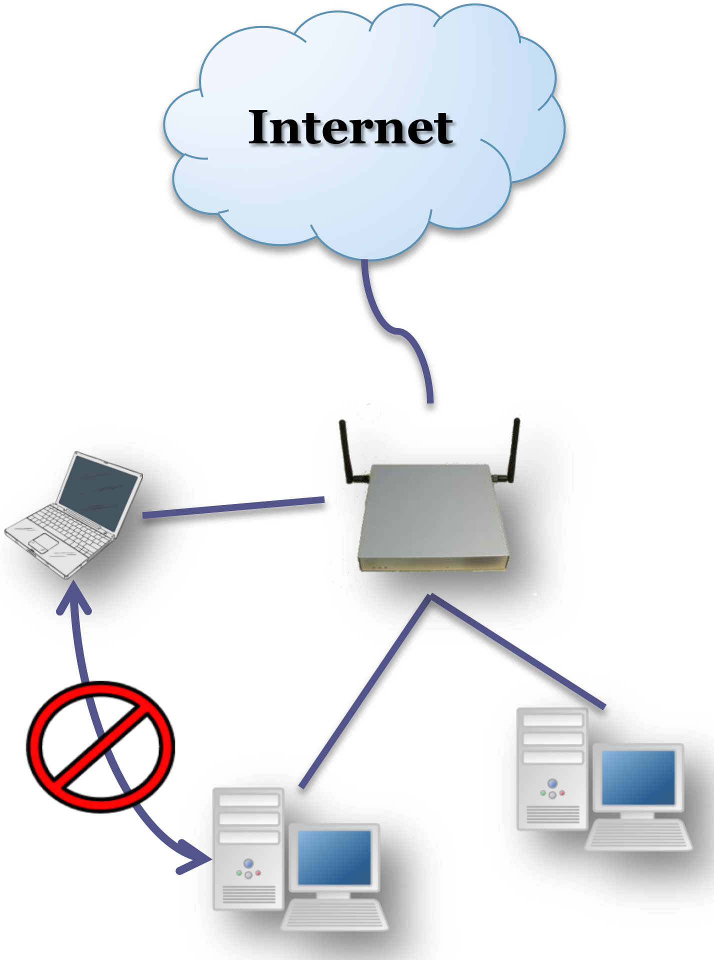
/Policies/UserManager/getUsers
/Policies/UserManager/addUser/guest
/Policies/UserManager/removeUser/Chris

Validation

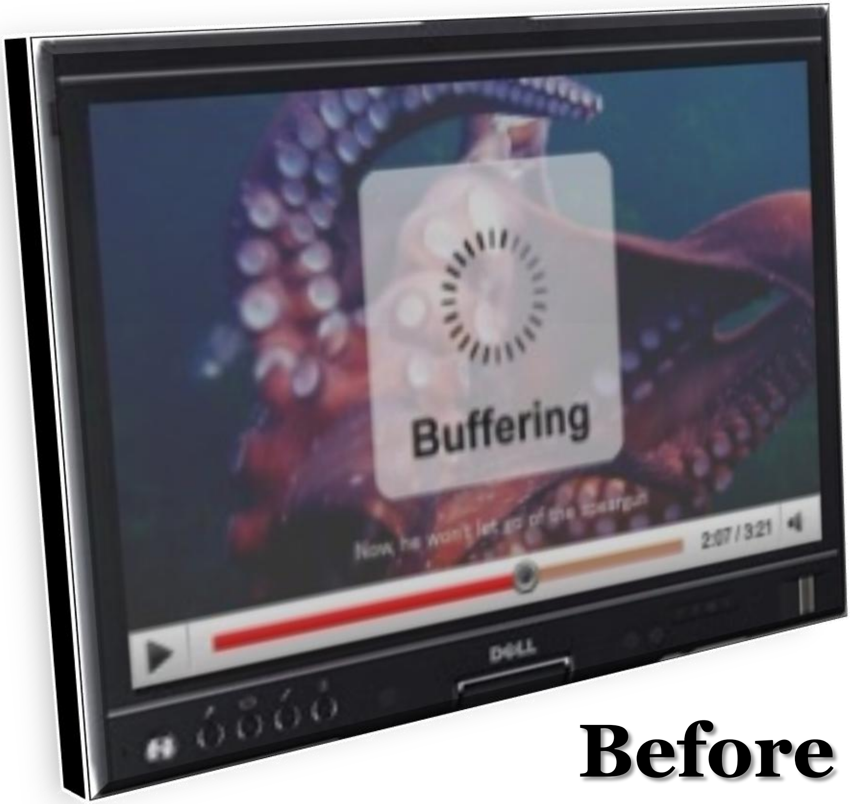
- Two Evidence Sources
  - Windows Messenger / MSN Messenger
  - Facebook
    - Gzip encoding
    - Large pool of servers (multiple IP addresses)
- Two Policies
  - Guest Isolation
  - Bandwidth Management

Guest Isolation

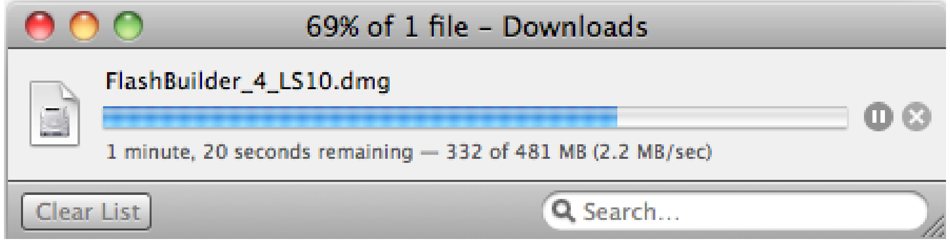




Bandwidth Management

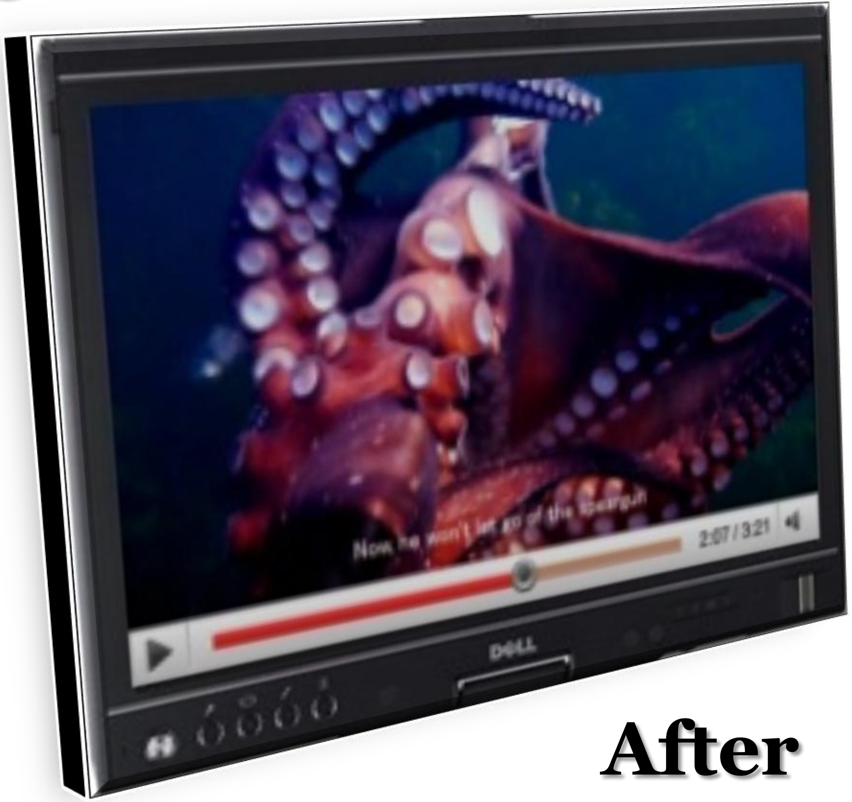


Before

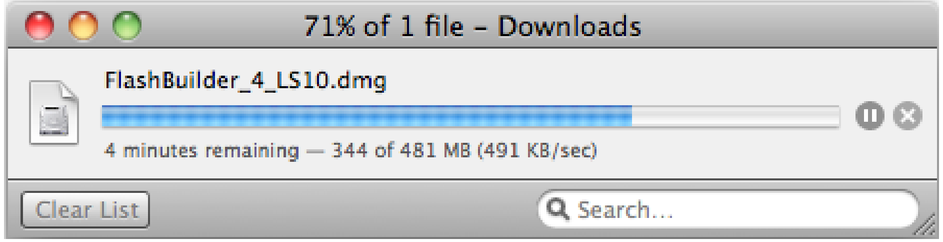

69% of 1 file – Downloads

FlashBuilder_4_LS10.dmg

1 minute, 20 seconds remaining — 332 of 481 MB (2.2 MB/sec)



After


71% of 1 file – Downloads

FlashBuilder_4_LS10.dmg

4 minutes remaining — 344 of 481 MB (491 KB/sec)

Conclusion

- A lot of architecture implemented
- Validated architecture with some applications
- Good start for an open development community

NOX
at HOME

Any Questions?

OpenFlow

NOX
An OpenFlow Controller