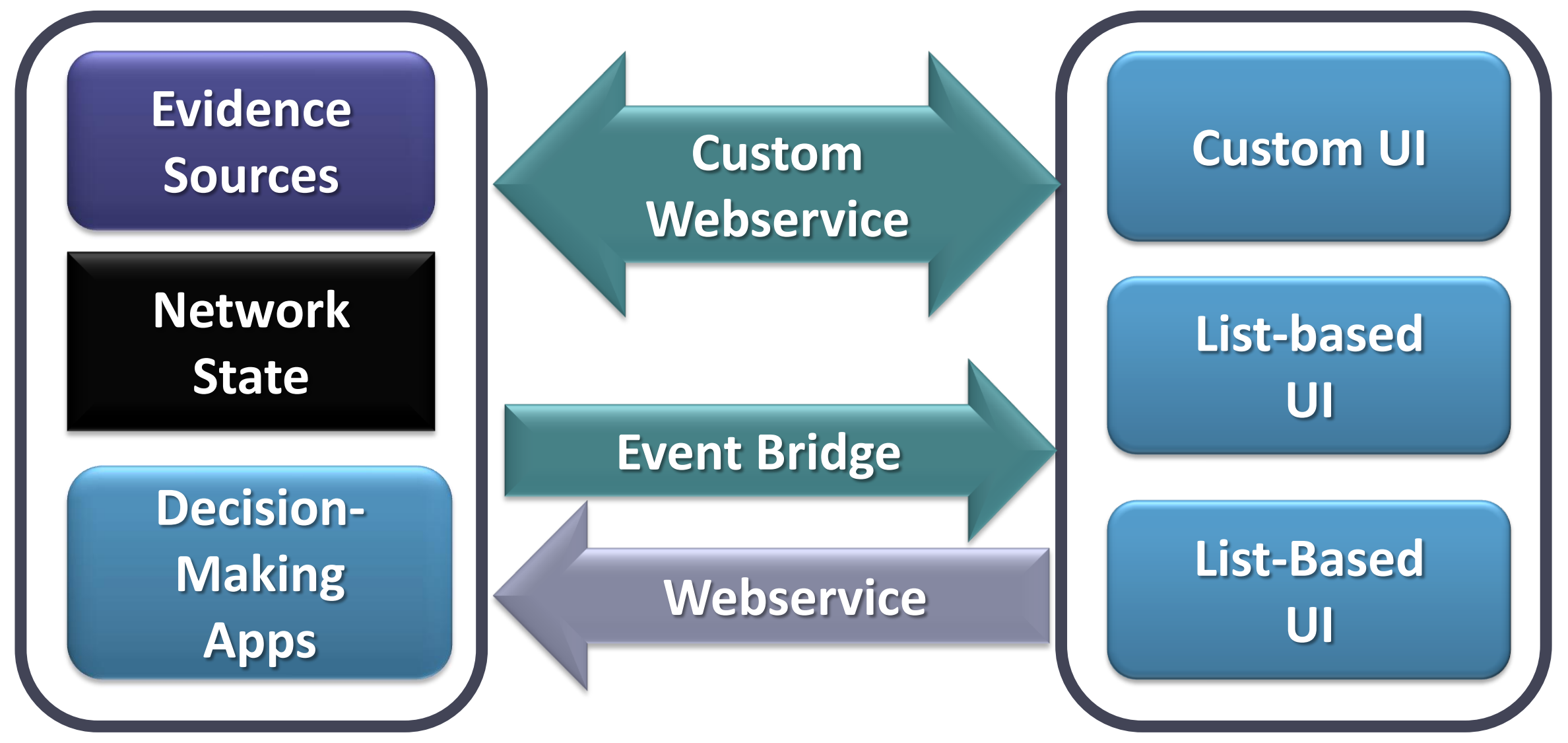
Decision Making in Home Networks

Andrew Arminio, Christopher Austin, James McCauley

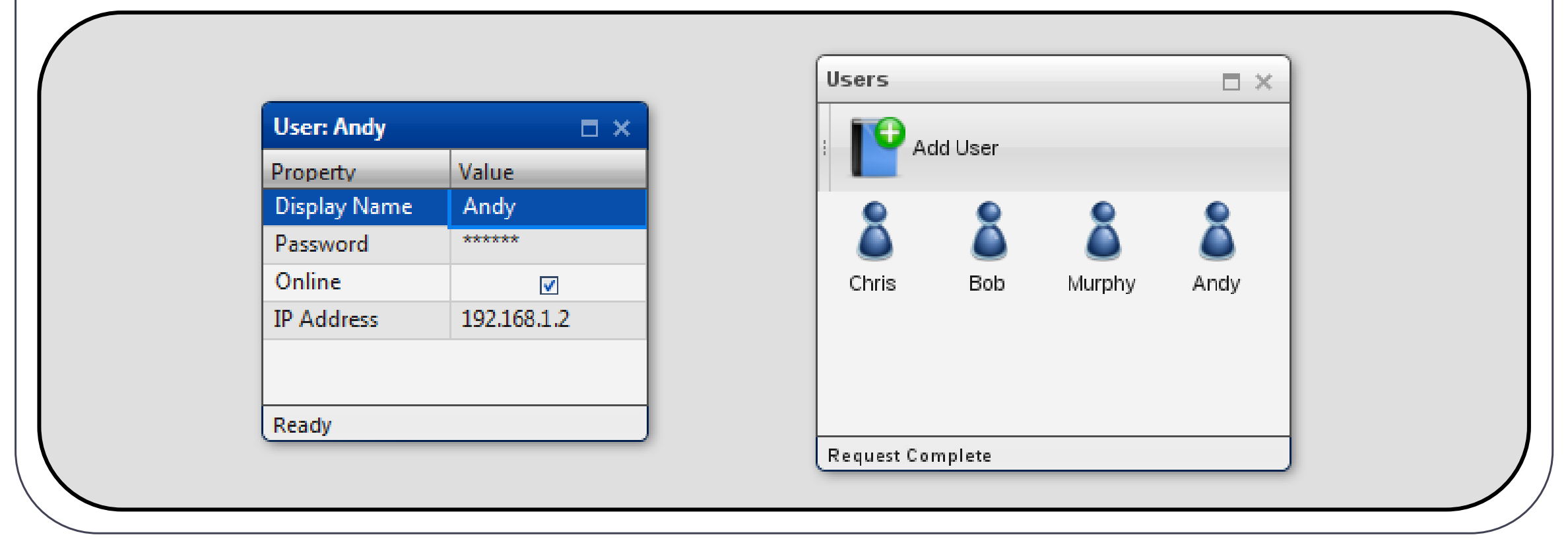
User Interface

Architecture



List-Based User Interface

- Many user interfaces consist primarily of lists (lists of users, lists of policies, list of configuration options, etc.)
- Our List-Based UI system creates a web UI based on a simple description of the list built using Python lists and dictionaries
- No HTML or JavaScript is required!



User: Andy

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

Ready

Users

Add User

Chris Bob Murphy Andy

Request Complete

Automatic Webservice Mapping

- NOX itself provides custom webservice creation
- The webservice mapping module automatically turns Python objects into webservices
- This allows developers to make already created functionality available to UIs "for free"

Event Bridge

- NOX has an event-based programming model
- It's easy to create listeners for these within NOX Python code
- Events are also interesting to user interfaces, but getting them from NOX to browser the UI required a lot of work:
  - Write a custom NOX component
  - Listen to events
  - Package events in JavaScript-friendly format
  - Expose events through a custom webservice
- The Event Bridge makes this easy - simply write a listener in JavaScript in the UI, and the Event Bridge takes care of making sure events get from the router to the browser

EGR Design

Our Client: Nicira Networks

Develops infrastructure technologies for large networks.
- Load balancing
- Client traffic isolation for multi-tenant datacenters
- Virtual machine migration
- Facilitate cloud computing

NOX & OpenFlow

Two technologies used by Nicira
- Provide a platform for writing control software for networks
- Provide a centralized programming model
- Provide a C++ and Python API

The Question

Nicira came to us with a question:
How can our technologies be applied to small networks?

Approaching the Question

Identified potential applications:
- Bandwidth control
- Isolation of guest users
- Better parental control
- Centralized network diagnostics
- Password protected file sharing

Identified common requirements:
- Web-based UI
- Users and user management
- Machines and machine management
- Passive user identification

The Answer

There are far more applications for NOX and OpenFlow in the home than any single developer would have time for or interest in pursuing

.. and we're pretty sure we haven't thought of some of the really good ones!

There's a development community waiting to happen

.. but it needs some infrastructure to facilitate independent components from independent developers to all work well together

Challenges for such a platform include:
- We want to make it easy for developers to implement usable user interfaces -- if it's difficult, they might not even bother! *(See left panel)*
- We need to provide components with information about the network state -- which users are online, and on which machines. *(See right panel)*
- Other requirements included:
  - We needed a packet classification system more in line with the idea of multiple independent components than the one NOX comes with
  - We needed a more flexible system for manipulating traffic than the module included with NOX

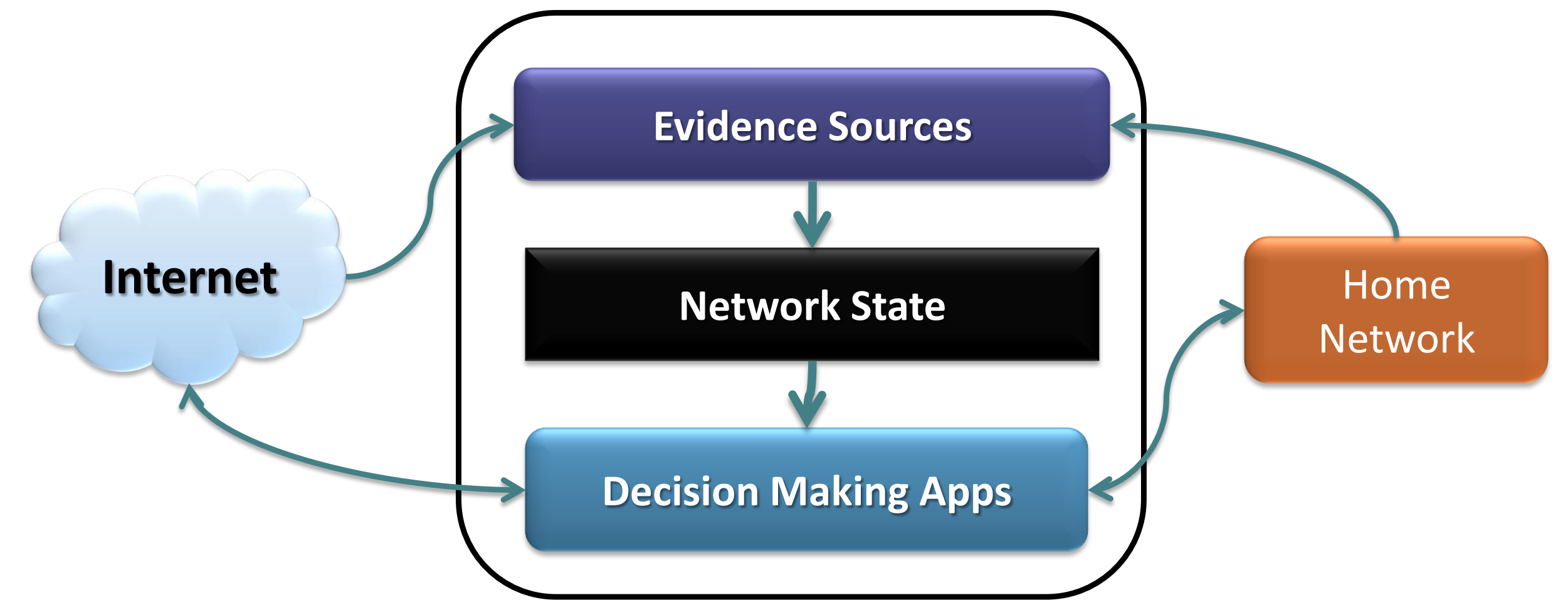
Where to go from here?

- Implement more evidence sources
  - iGoogle
  - Twitter
  - Yahoo IM
  - gTalk
  - World of Warcraft?
- Implement more complex evidence combination
- Extend and refine our existing policy apps (e.g. user-based throttling criteria)
- Apply list-based UI paradigm to form creation

.. get a community started!

Producing Functionality

Architecture



Passive User Identification

- Decision making components require information about the state of the network, especially users
- In large networks, NOX can rely on existing authentication mechanisms (802.1X, NT Domains, etc.) which do not exist in the home
- Rather than require explicit login, we support passive user identification
  - Evidence Source components provide the Network State component with evidence that a user is online
  - Many of these Evidence Sources work by monitoring existing internet identities (instant messaging logins, social networking logins, etc.)
  - Network State component combines these evidence messages to form a view of which users are online
  - Individual Decision Making Apps query the Network State

Evidence Sources

- Key to passive user identification
- Generally work by monitoring traffic

Network State

- Gathers evidence from Evidence Sources
- Makes determinations about Network State
- Provides state information to Decision Makers

Decision Making

- Will often implement a policy
- Generally:
  - Associates traffic with a user or machine by querying Network View
  - Controls traffic to carry out a policy

NORTHERN ARIZONA UNIVERSITY
College of Engineering, Forestry & Natural Sciences