

# Making a Mesh of the Infrastructure

## The DRPful Approach

Pete Brown

Network Engineer, AutoZone

@jpbrown1





# Pete Brown

Network Engineer, AutoZone

Pete has supported various functional areas of infrastructure for around 20 years. He is a proponent of democratizing access to infrastructure data in order to promote learning, facilitate better decision making and reduce reliance on tribal knowledge.

# The Challenge

We are often asked to collect and analyze data from disparate infrastructure source types. These efforts are hindered by constant technological and environmental changes.

How can we mitigate the impact of these changes to reduce time spent on development and maintenance?

# Terminology

## Infrastructure System

Any system normally managed by IT groups and required by users or applications for day to day operations. This includes network components, directory services, collaboration services, identity management and others.

## Source

Any system from which data needs to be retrieved. Since some infrastructure systems do not offer APIs, the term API would not suffice. These systems require the use of protocols such as LDAP, SNMP, various flavors of SQL, etc.



So... many.. sources...

# Sample Sources

Wireless  
Mgmt

Telephony

Directory  
Services

IPAM

Load  
Balancers

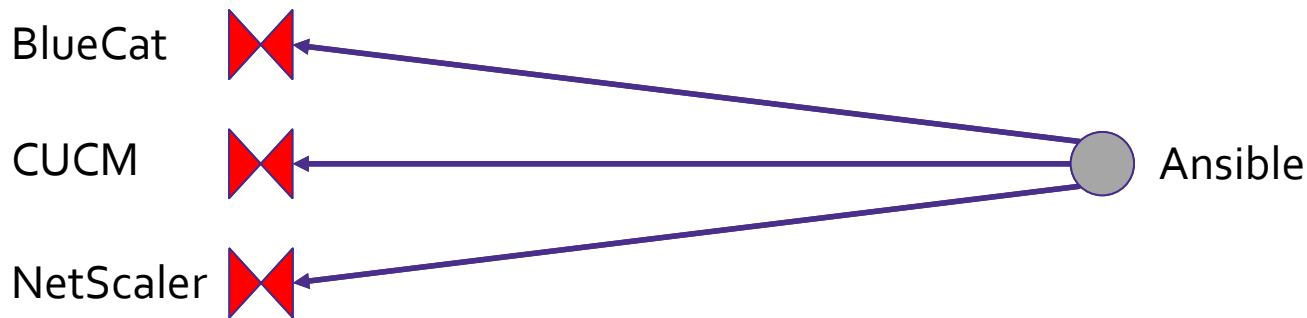
Identity  
Mgmt

Hardware  
Inventory

# Sample Sources

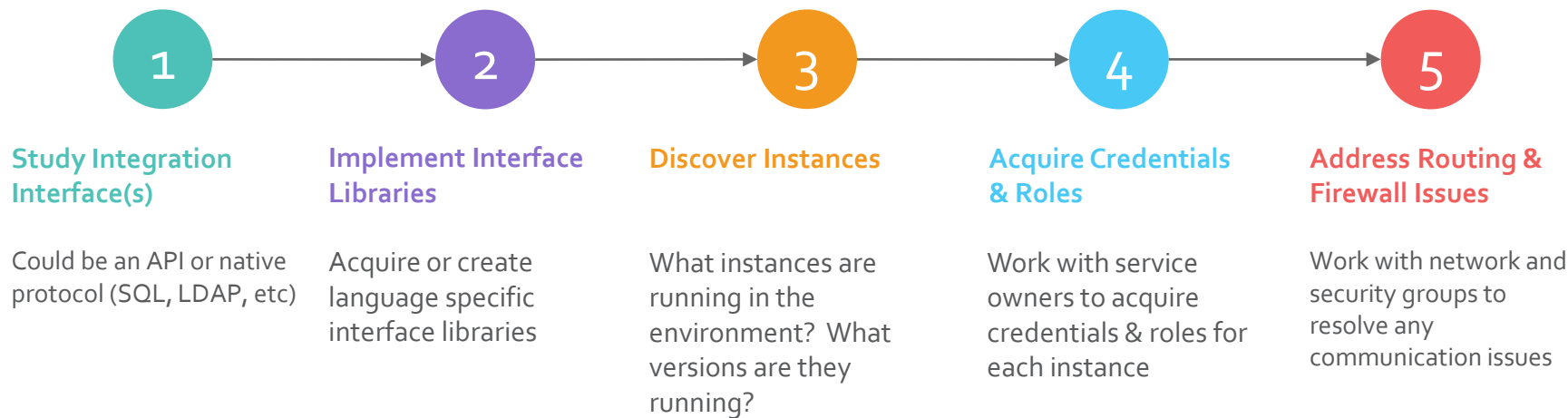
Source Type	Protocol	Data Format	Spec Format	Spec Source	Client Language
Meraki	REST	XML/JSON	OpenAPI	Application	any
CUCM – AXL/RIS	SOAP	XML	PDF/HTML	Application	any
CUCM - JTAPI	JTAPI	binary	PDF/HTML	Application	Java
CUC - CUPI	REST	XML	PDF/HTML	Internet	any
NetScaler	REST	JSON	PDF/HTML	Internet	any
Active Directory	LDAP	binary	PDF/HTML	Internet	any

# Accessing a Source - Traditional





# Accessing a Source - Traditional

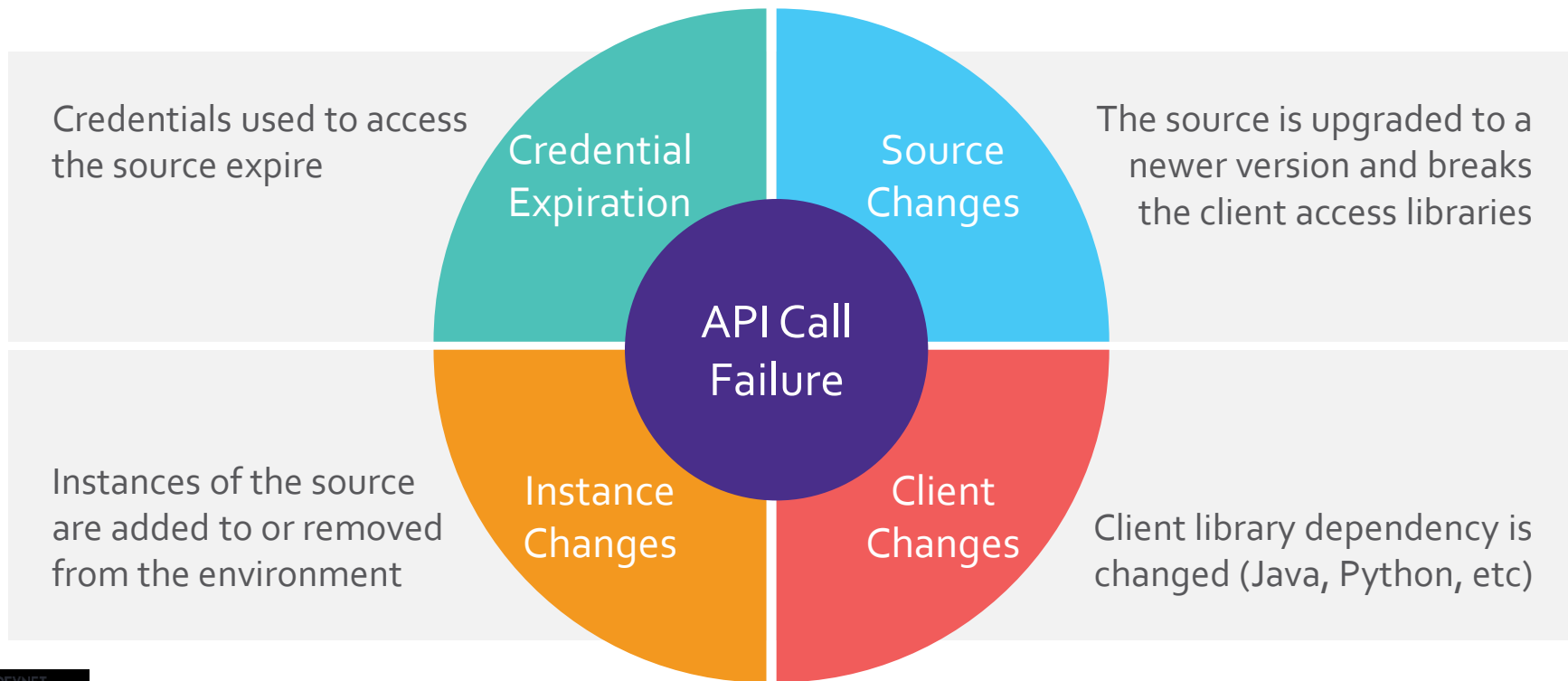


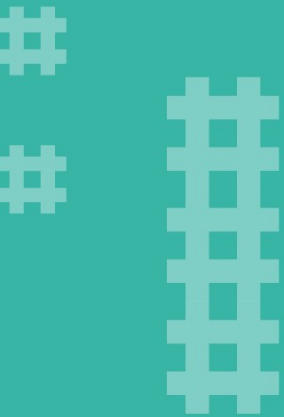
**Great! You've integrated directly  
with all the things. All done.**

... but wait, there's more...

8 months later your phone rings at 2am  
You have entered...

# The API Circle of Suffering





# Enter the Infrastructure Mesh

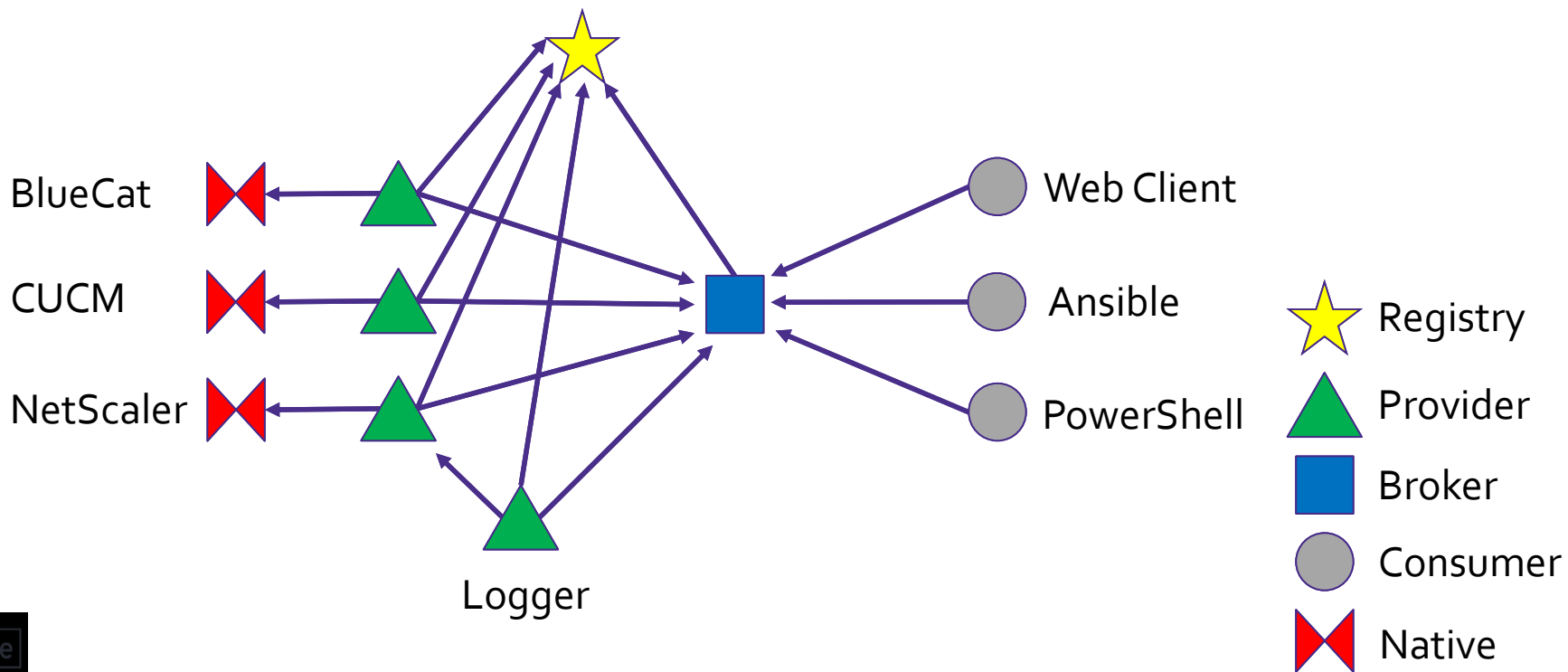
# DRP – Declarative Resource Protocol

- A JSON based WebSocket subprotocol for declaring and consuming resources
- Provides a relatively easy way to create a service mesh for the infrastructure
- Allows consumers to focus on data analysis functions by reducing time spent on discovery and connectivity
- The Registry and Broker nodes are meant to run as part of the network (containers on IOS XE devices)

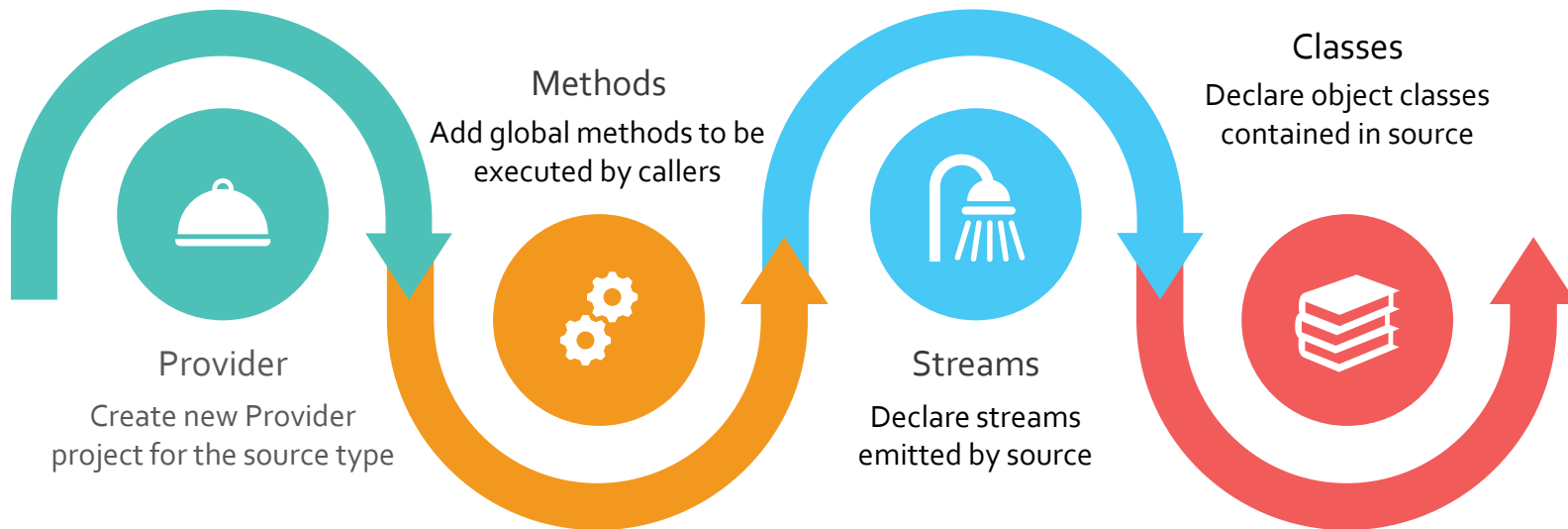
# DRP – Declarative Resource Protocol

- Sources are declared, not discovered
- Sources use a common format for declarations
- Consumers use a single logical endpoint to access all sources
- Consumers use a standardized RPC & pub/sub mechanism to access all sources
- Promotes an integrate once, use many approach

# DRP Mesh Components

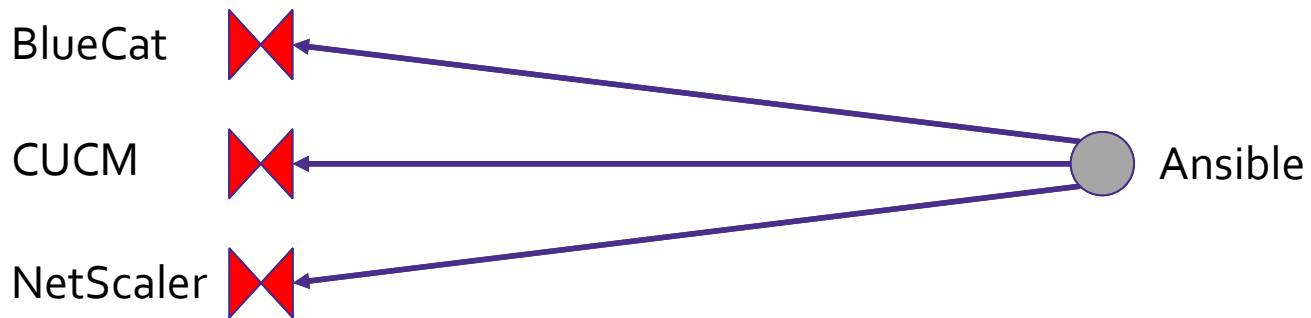


# Creating a Provider Type

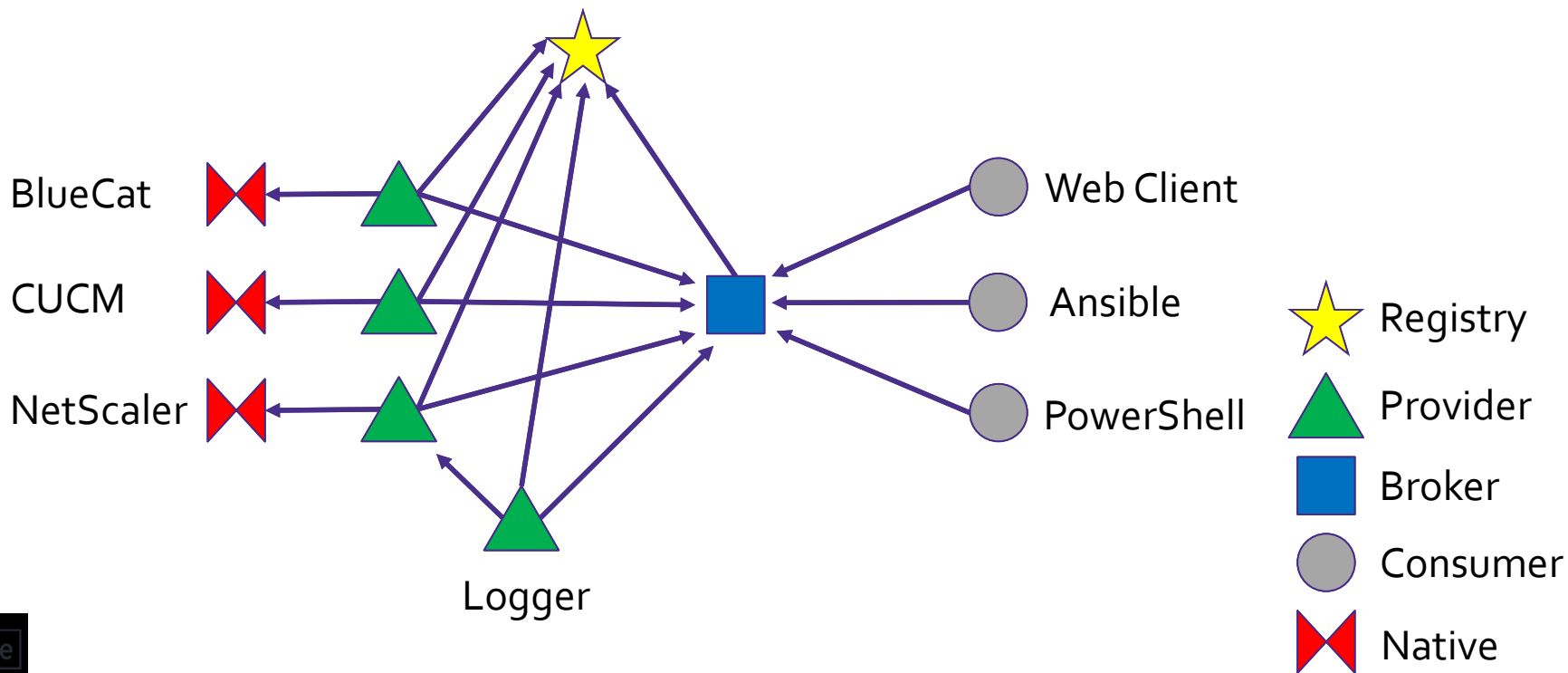




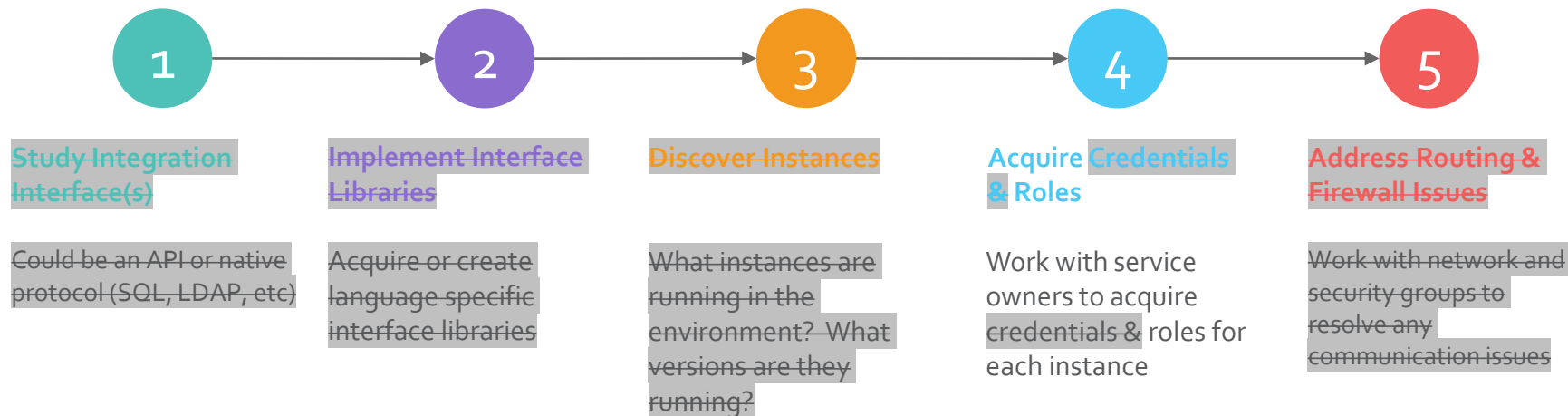
# Accessing a Source - Traditional



# Accessing a Source - Mesh



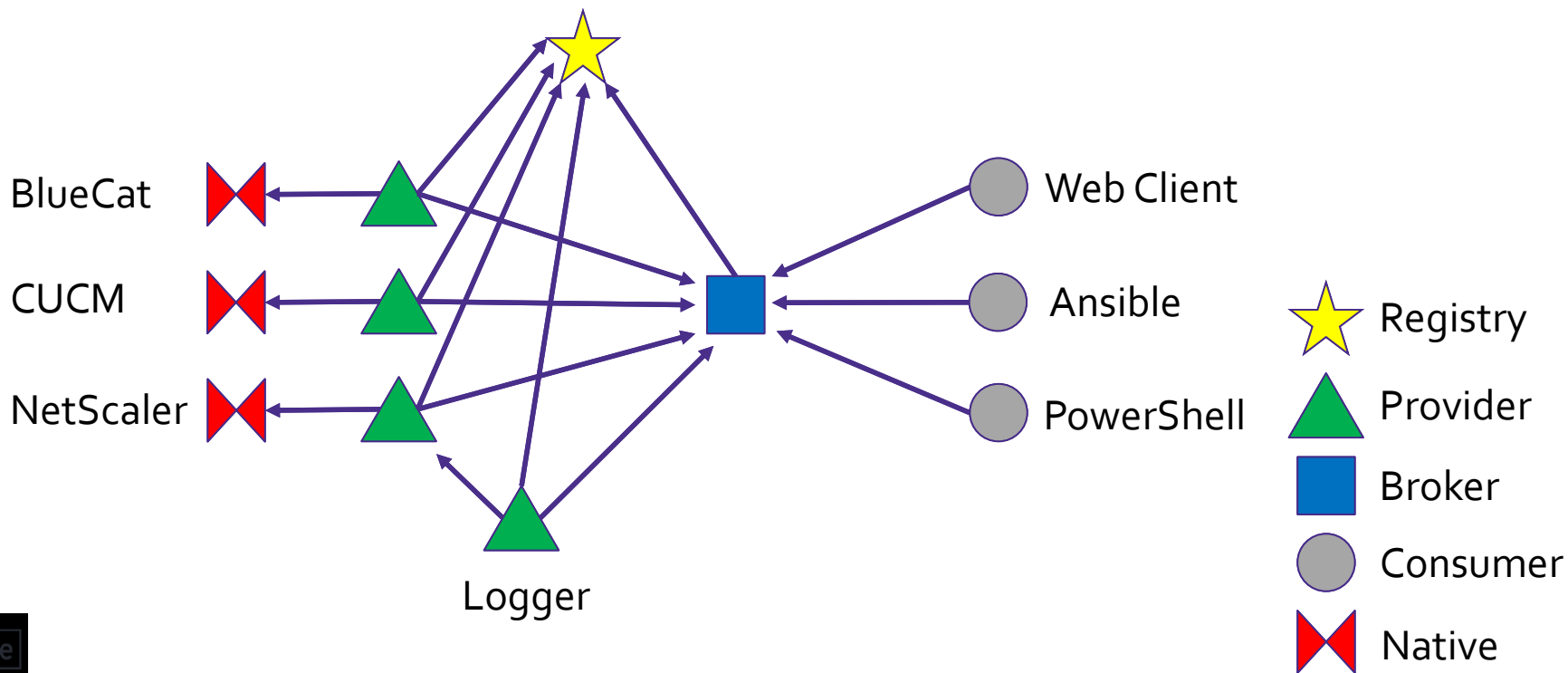
# Accessing a Source – Mesh





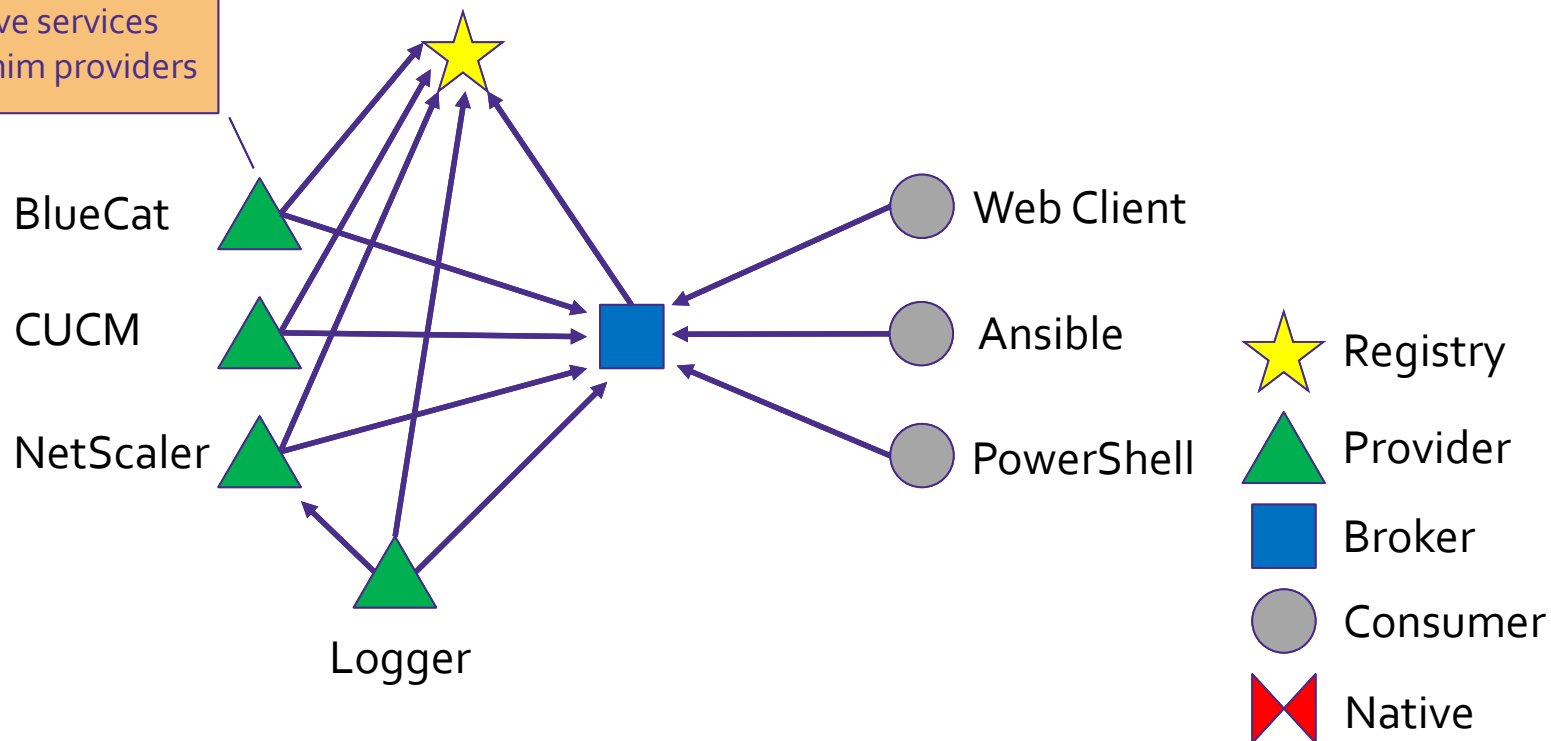
# Demo Time!

# DRP Mesh Components



# DRP Mesh Components – Future?

DRP capable native services  
would not require shim providers



# Is it for you?

## Pros

- Simplifies source access
- Promotes integrate once, use many approach
- Promotes learning
- Allows the use of non-listening services
- Can inject services into insecure zones
- Potentially consolidates access to all aspects of an object (CUCM AXL/RIS/JTAPI)
- Reduces perception of “black boxes”
- Can enhance HA/DR capabilities
- Optional caching mechanism
- Opens door to using GraphQL with disparate infrastructure sources

## Cons

- Loss of resolution in source system logs (single API user) for pre-existing sources
- Elimination of shortest route between consumer and source
- Dynamic nature of connections may cause confusion when troubleshooting
- Small development team (2)

# Thank you for coming!

Project used in demo:

<https://github.com/adhdtech/DRP>





DEVNET  
Create