Introducing Pliny

Justin Ross

6 January 2014

What is it?

Pliny is a web application for storing and provisioning AMQP queues and topics. It's for use by organizations that want to offer self-service messaging to developers of distributed applications.

- Users can create and save AMQP node definitions
- Users can provision network resources for their addresses
- Pliny gives each user her own messaging namespace

Addresses: Pliny's plan for addresses

Pliny's addresses impose a little more structure than AMQP addressing by itself requires.

```
//$domain/$authority/$party/$arbitary
```

- `$domain` is the DNS name of an AMQP messaging endpoint
- `$authority` is the name of the Pliny deployment
- `$party` is the name of a user or group
- `$arbitrary` is something you choose for your application

Addresses: A domain for connections

Every address has a DNS name for the messaging broker or router that provides the messaging service.

```
//$domain/$authority/$party/$arbitary
  ^^^^^^^

//router.example.net/$authority/$party/$arbitrary
  ^^^^^^^^^^^^^^^^^^

//bus.apache.org/$authority/$party/$arbitrary
  ^^^^^^^^^^^^^^
```

This is the information clients need to make a network connection.

Addresses: A Pliny deployment is the authority

All addresses from a particular Pliny deployment share one `$authority`, a key for all the addresses under a single Pliny instance.

```
//$domain/$authority/$party/$arbitary
          ^^^^^^^^^^

//router.example.net/demo/$party/$arbitrary
                     ^^^^

//bus.apache.org/test/$party/$arbitrary
                 ^^^^

//bus.apache.org/prod/$party/$arbitrary
                 ^^^^
```

An organization can deploy multiple Pliny instances. The `$authority` ensures that addresses from these instances don't collide. And each Pliny instance ensures that no addresses under its authority collide.

Addresses: You have your own namespace

All pliny addresses are namespaced by the party they belong to.

```
//$domain/$authority/$party/$arbitary
                     ^^^^^^

//router.example.net/demo/humphrey/$arbitrary
                          ^^^^^^^^

//bus.apache.org/prod/infra/$arbitrary
                      ^^^^^
```

A party is a user or a group: “humphrey” or “josephine”, “sales” or “engineering”.

Because Pliny addresses are always namespaced in this fashion, they can freely move between public and private access without danger of collisions.

Addresses: The rest is arbitrary

After `$party`, the address is left to the discretion of the user. Choose something that makes sense for your application.

```
//$domain/$authority/$party/$arbitary
                            ^^^^^^^^^

//router.example.net/demo/humphrey/news/europe
                                   ^^^^^^^^^^^

//router.example.net/demo/humphrey/news/asia
                                   ^^^^^^^^^

//bus.apache.org/prod/infra/builds.x86_64
                            ^^^^^^^^^^^^^

//bus.apache.org/prod/infra/commits
                            ^^^^^^^
```

Address configuration

In addition to the address name, the user controls a few other important address policies.

- *Private or public* - A private address is accessible only to clients with correct party credentials
- *Distribution mode* - Policy for passing messages out to receivers
  - *Single receiver* - Only one receiver at a time is allowed
  - *Fan out* - Give each receiver a copy of every message
  - *Round robin* - Balance message deliveries evenly among receivers
  - *Lowest cost* - Give priority to closer receivers

Provisioning

Pliny isn't responsible for sending and receiving messages. It only holds information about addresses. To make an address useful to an application, something must take that information and use it to provide a messaging service.

Pliny communicates with (well, at this point, *intends* to communicate with) Qpid Dispatch, telling it what addresses exist under the domain of a given Pliny deployment.

Management and monitoring

The scope of address management and monitoring in Pliny is not yet defined.

Let's try it

Begin demo

More information

Source code

https://github.com/ssorj/pliny