BLUE BRAIN PROJECT

ZeroEQ

Zero Event Queue — Easy, safe, fast messaging for C++

What is ZeroEQ?

BLUE BRAIN PROJECT

- C++ library for event-driven distributed architectures
  - ZeroMQ: robust and efficient data transport
  - Zeroconf: automatic resource discovery
  - ZeroBuf: Efficient, strong typed, strong semantic Serializables
- Push-based event propagation
  - Subscribe to events, receive them when they appear
  - Loose coupling to event publishers
- HTTP GET/PUT server with JSON payload

Example: Publish-Subscribe

BLUE BRAIN PROJECT

Emitting application

```
zerobuf::render::Camera camera;

zeq::Publisher publisher;

while( true )
    publisher.publish( camera );
```

Receiving application

```
zerobuf::render::Camera camera;

zeq::Subscriber subscriber;

subscriber.subscribe( camera );
subscriber.receive();
```

- Zeroconf auto-subscribes within the same session (=user)
  - zeq::Subscriber(URI) for explicit subscription
- Receive blocks for one event

Example: HTTP Server

BLUE BRAIN PROJECT

HTTP Server

```
zerobuf::render::Camera camera;

zeq::http::Server server( ”:4020” );
server.add( camera );

while( true )
  server.receive();
```

Web Client

```
> telnet localhost 4020
Escape character is '^]'.
GET /zerobuf/render/Camera HTTP/1.1

HTTP/1.0 200 OK
Content-Length: 197

{
   "lookAt" : {
      "x" : 0,
      "y" : 0,
      "z" : 0
   },
   "origin" : {
[…]
```

Features

BLUE BRAIN PROJECT

- Publisher
  - Publish objects with simple Serializable interface
  - ZeroBuf generates Serializable objects
- Subscriber
  - Register Serializable objects
  - Updated upon receive()
  - Serializable has update notification function

Features

- HTTP Server
  - Synchronous due to HTTP design
  - Behaves similar to a combined publisher+subscriber
  - GET for registered objects
  - PUT for subscribed objects
  - JSON payload

Use Cases

BLUE BRAIN PROJECT

- Loosely couple similar applications
  - Camera, selection, time, simulation results, …
- Bridge C++ applications into web services
- Remotely execute components of an application
  - Computation, IO, …
- Build distributed data processing applications
- Decouple GUI from application
  - http::Server + Javascript

To Do

BLUE BRAIN PROJECT

- GET /object/schema for http::Server
- Management server
  - Query schema, state and subscriptions
  - Initiate subscribe to a concrete publisher and event
  - For manual configuration through a “nodegraph” GUI