

# Notificaciones a través de SocketIO y ZeroMQ

Arturo Nikolai Sevilla  
Covarrubias  
David Esp(eran|ar)za  
Hernández

# Problema

- Envío de notificaciones a usuarios suscritos.
- Sistema desacoplado y Escalable.

# Solución

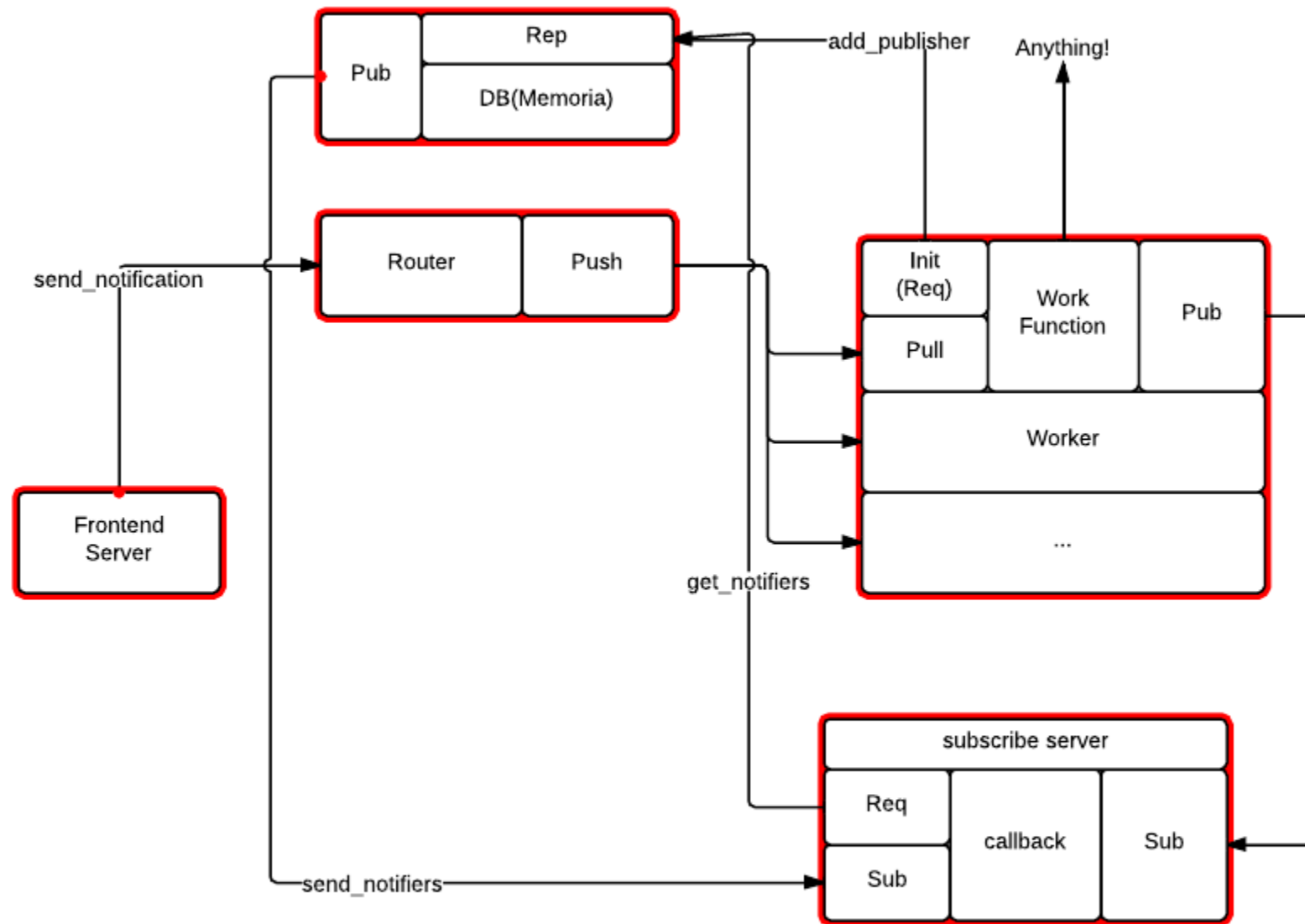
- SocketIO – conexión Nodejs - cliente.
- Nodejs – solicitudes web y subscripciones.
- ZeroMQ – notificadoros y publicaciones.
- Redis – información de sesión.

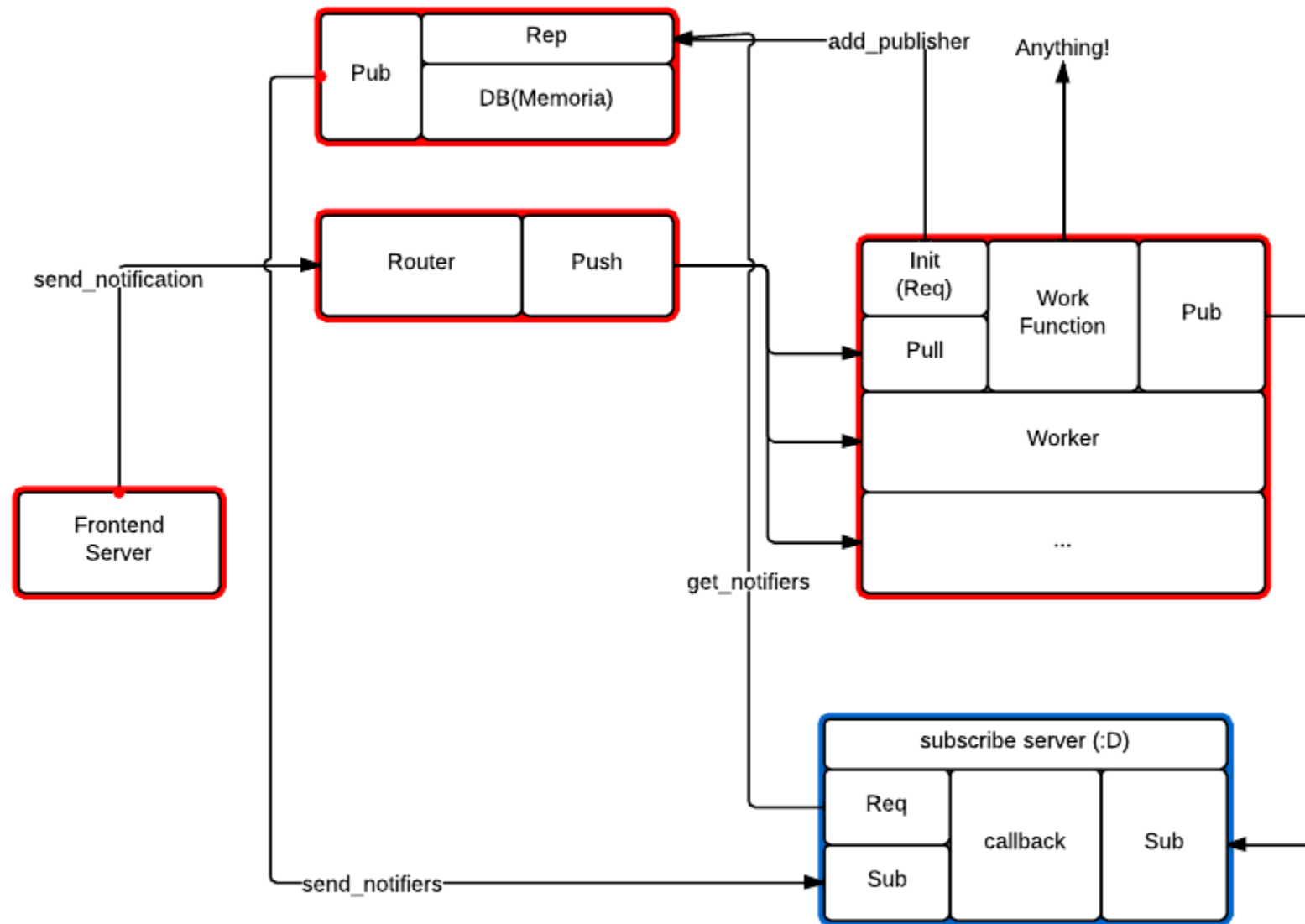
# Nodejs

- Concurrencia.
- Fácil de utilizar
  - Ambiente
  - Lenguaje
- SocketIO.

# Redis

- Manejo de sesiones.
- Velocidad.
  - Memoria





# Nodejs

- Dependencias:
  - socket.io
  - pickle
  - zmq
  - Redis
- Elementos:
  - server.js
  - notificationpipe.js
  - user.js



# notificationpipe.js

Bindings a ZeroMQ.

Flujo:

- Traer los notificadores.

```
pubNotifierServer.connect(reqServerConfig);
for (var x in notAddresses) {
  addPublisher(notAddresses[x]);
}
```
- Suscribirse.

```
subscribeToUserNotifications(user);
```
- Escuchar.

```
notifierSocket.on('message', function(message) {});
var registrations = registeredUsers[recipient];
for (var x in registrations) {
  registrations[x].callback(notificationObject);
}
```

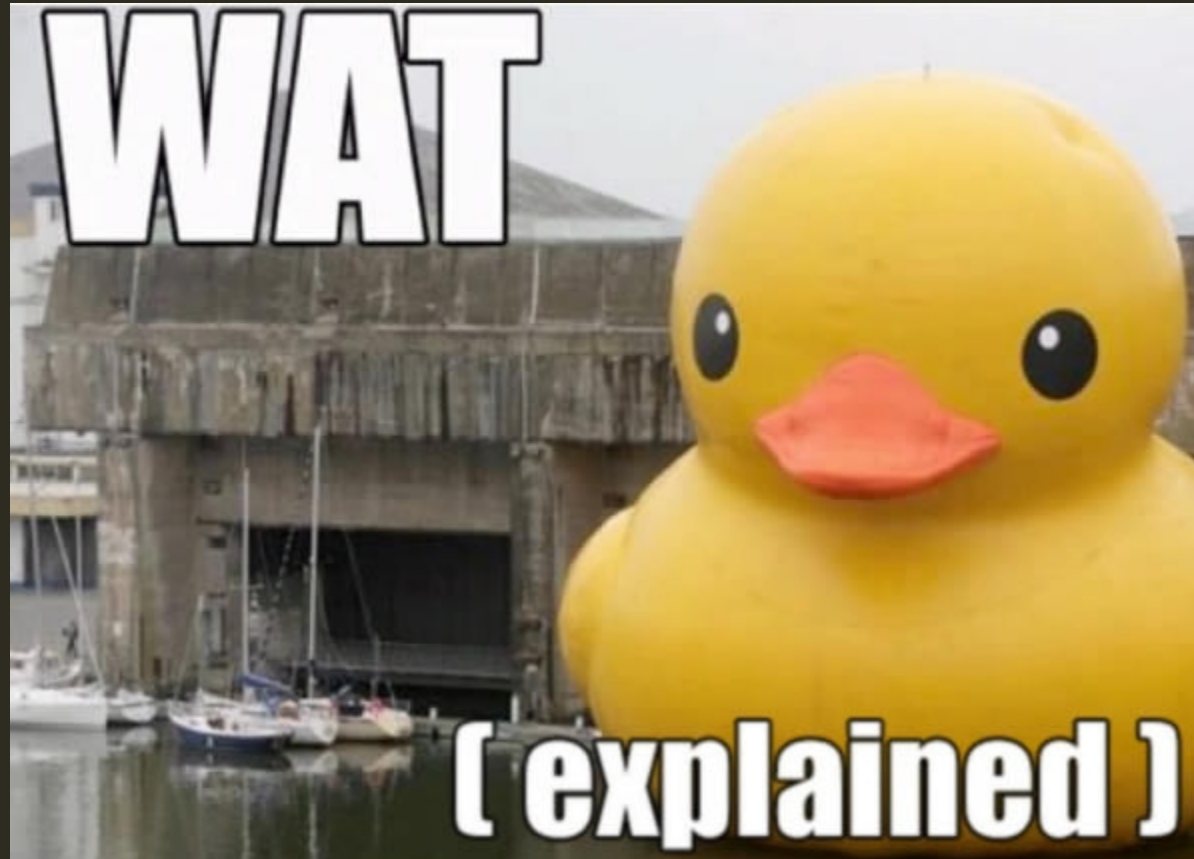
# user.js

Bindings al servidor Redis.

- Información de sesiones (sessionId).
- Recuperar información de Usuario.

```
var redisClient = redis.createClient(redisPort, redisHost);  
redisClient.get(key, callback(err, reply));
```

Demo time



# Mejoras

- Debido a que es un sistema originado de un proyecto alternativo está muy acoplado a dicho proyecto.
  - Repoze.who.tkt
  - Beaker
- Seguridad en el sistema de suscriptores.
- Testing!!