Gossip XMPP Proxy

Grupo 3

Federico Bond - 52247

María de la Puerta E. - 50009

Alexis Medvedeff - 50066

Juan Pablo Rey - 50265

ITBA
INSTITUTO TECNOLOGICO DE BUENOS AIRES
·UNIVERSIDAD PRIVADA·

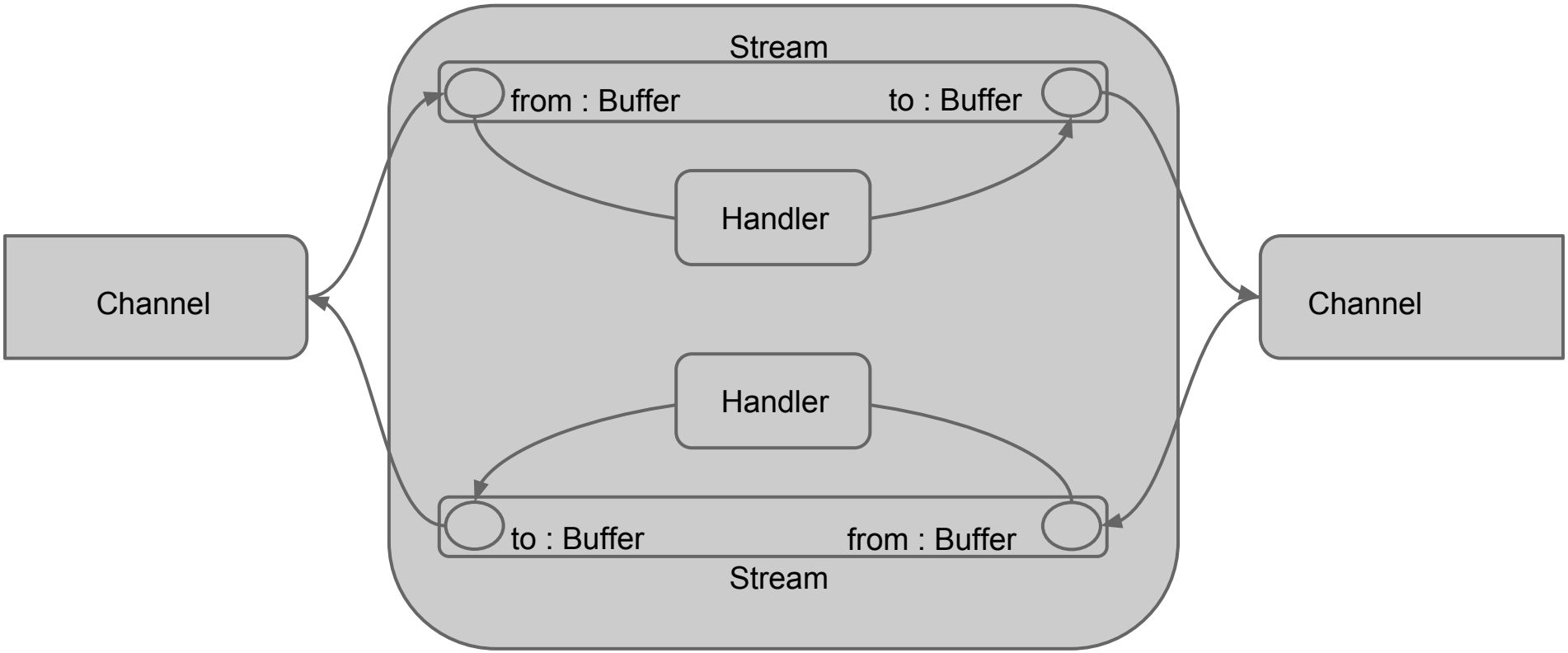
Agenda

- Arquitectura general
- Librerías utilizadas
- Protocolo de Administración
- Dificultades y posibles mejoras
- Conclusiones

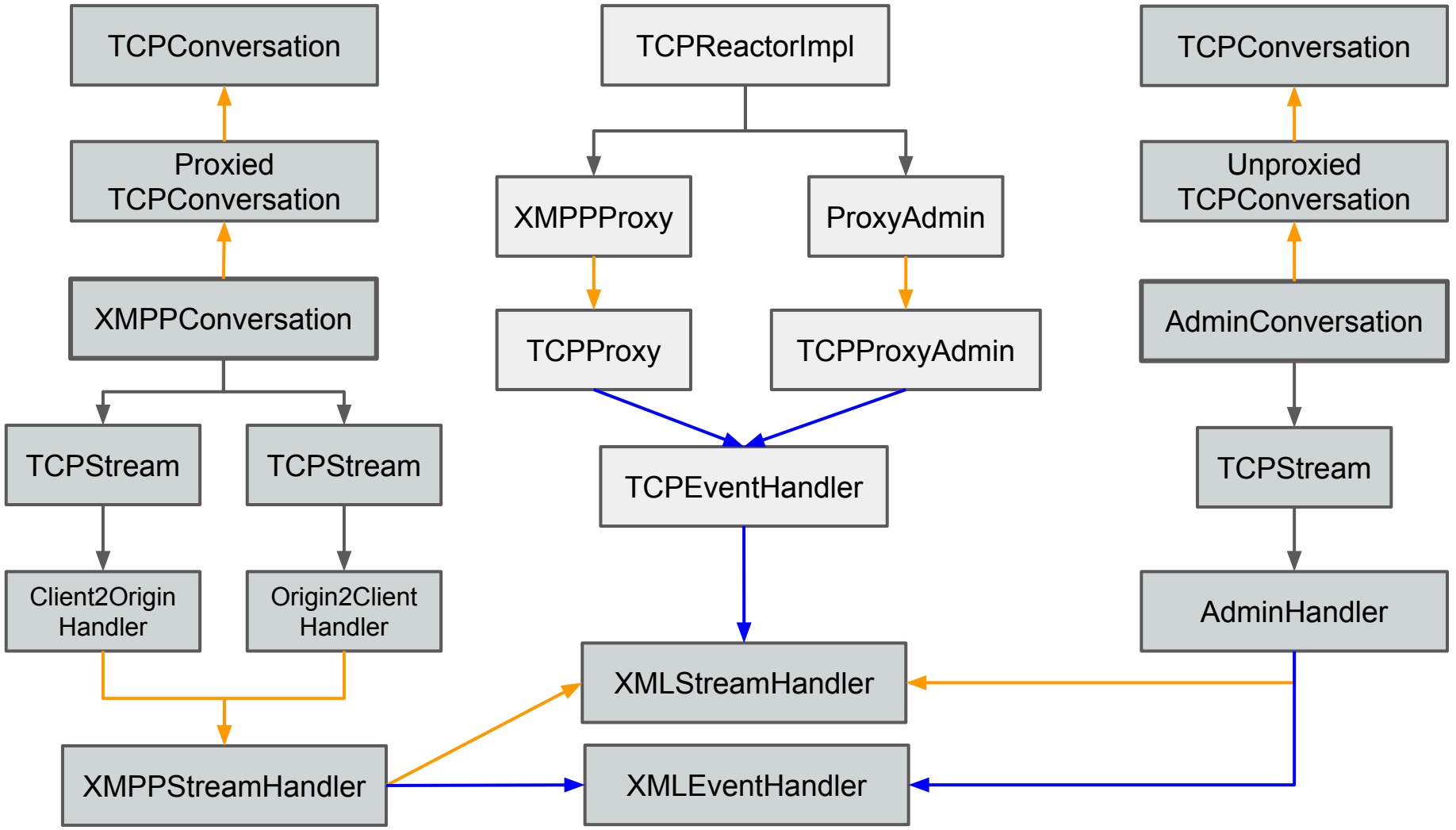
Arquitectura General

- Capas
  - TCP
  - XML
  - XMPP
- Componentes
  - Proxy
  - Admin

Arquitectura General
Comunicación del proxy



Arquitectura General



Patrón Reactor

Maneja eventos concurrentes originados por múltiples clientes.

Demultiplexa los pedidos entrantes y los entrega de forma sincrónica.

- Recursos de entrada/salida
- Despachador
- Manejador de pedidos

SI File Transfer

Soporta dos mecanismos

- In-band streams
- SOCKS5 streams

Librerías utilizadas

- Aalto XML - Parsing no bloqueante de streams XML
- Apache Commons - Codificador Base64 y utilidades varias
- JUnit / Mockito - Testing
- SLF4J - Logging

Estadísticas de uso

1. Conexiones al proxy
2. Bytes leídos
3. Bytes escritos
4. Mensajes XMPP enviados por clientes
5. Mensajes XMPP recibidos por clientes

Acceso mediante protocolo de administración

Protocolo de Administración

- Protocolo basado en XML.
- Se debe conectar al puerto configurado para administración.
- La conversación es iniciada por el cliente con la declaración de XML: `<?xml version=”1.0”?>`.
- Debe enviarse el tag raíz `<admin>` para luego enviar comandos.

Protocolo de Administración
Comandos

Comandos del administrador	Respuestas
<usr>*username*</usr> <pass>*password*</pass> *<leet>on|off</leet>* <stats>*1|2|3|4|5*</stats> <silence value=”*on|off*”>*JID*</silence> <origin usr="*username*">*address*</origin> <quit/>	<success/> <error/> <error> <code>*error code*</code> <message>*error message*</message> </error> <stats> <type>*1|2|3|4|5*</type> <desc>*stat description*</desc> <value>*stat value*</value> </stats>

Protocolo de Administración Ejemplo

<?xml version="1.0"?> <admin> <usr>admin</usr>	
	<success/>
<pass>1234</pass>	
	<success/>
<stats>4</stats>	
	<stats> <type>4</type> <desc>Number of messages from clients</desc> <value>35</value> </stats>
<quit/>	
	<success/>

Dificultades encontradas

- Arquitectura **compleja**: aportó mucha claridad a la hora de agregar funcionalidad pero ante cambios ortogonales se pierde fácilmente la perspectiva

- Dificultades a la hora de hacer tests (tanto unitarios como de carga) por la complejidad del problema

Posibles mejoras

- Protocolo de administración basado en estándares XMPP
  XEP-0050: Ad-Hoc Commands
- Reactor multi-threaded
- Implementar mecanismos seguros de conexión

Conclusión

- Ganamos experiencia en el desarrollo de aplicaciones de red concurrentes
- Aprendimos sobre protocolos ampliamente utilizados como XMPP
- Apreciamos los desafíos que involucra desarrollar aplicaciones de alto rendimiento para sistemas en red