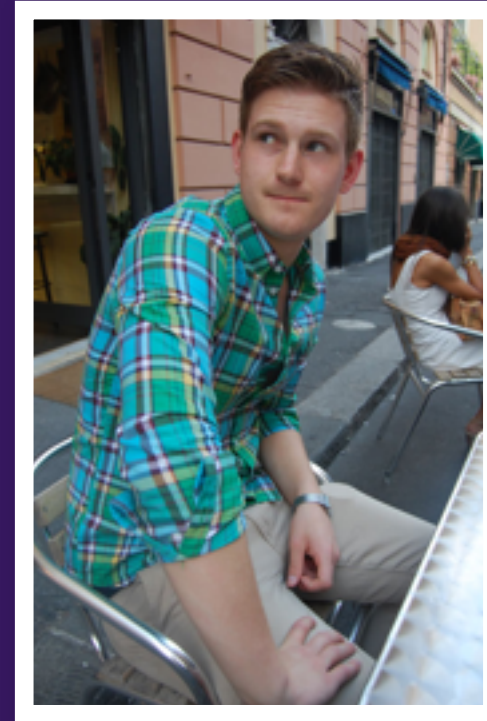


# The (Almost) Realtime Web

Why you need to care about WebSockets



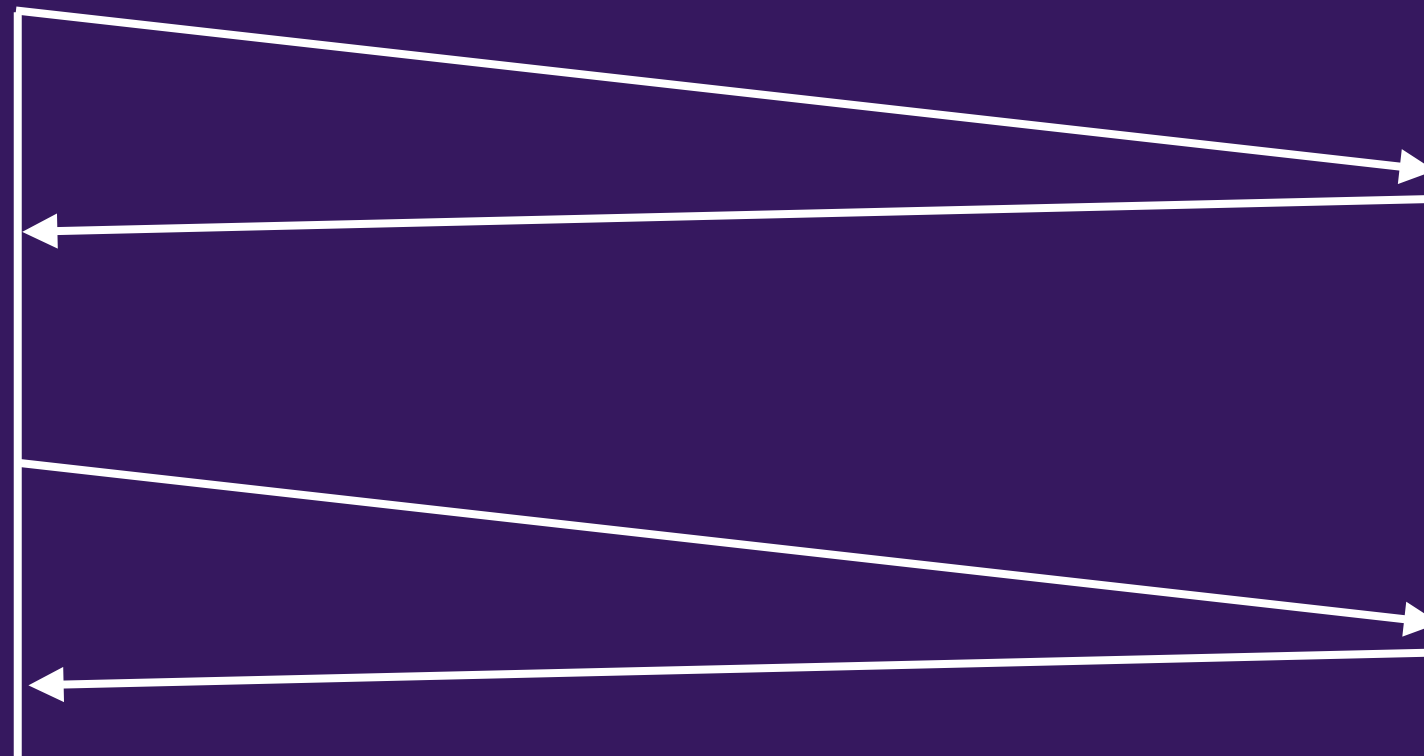
Calle Wester



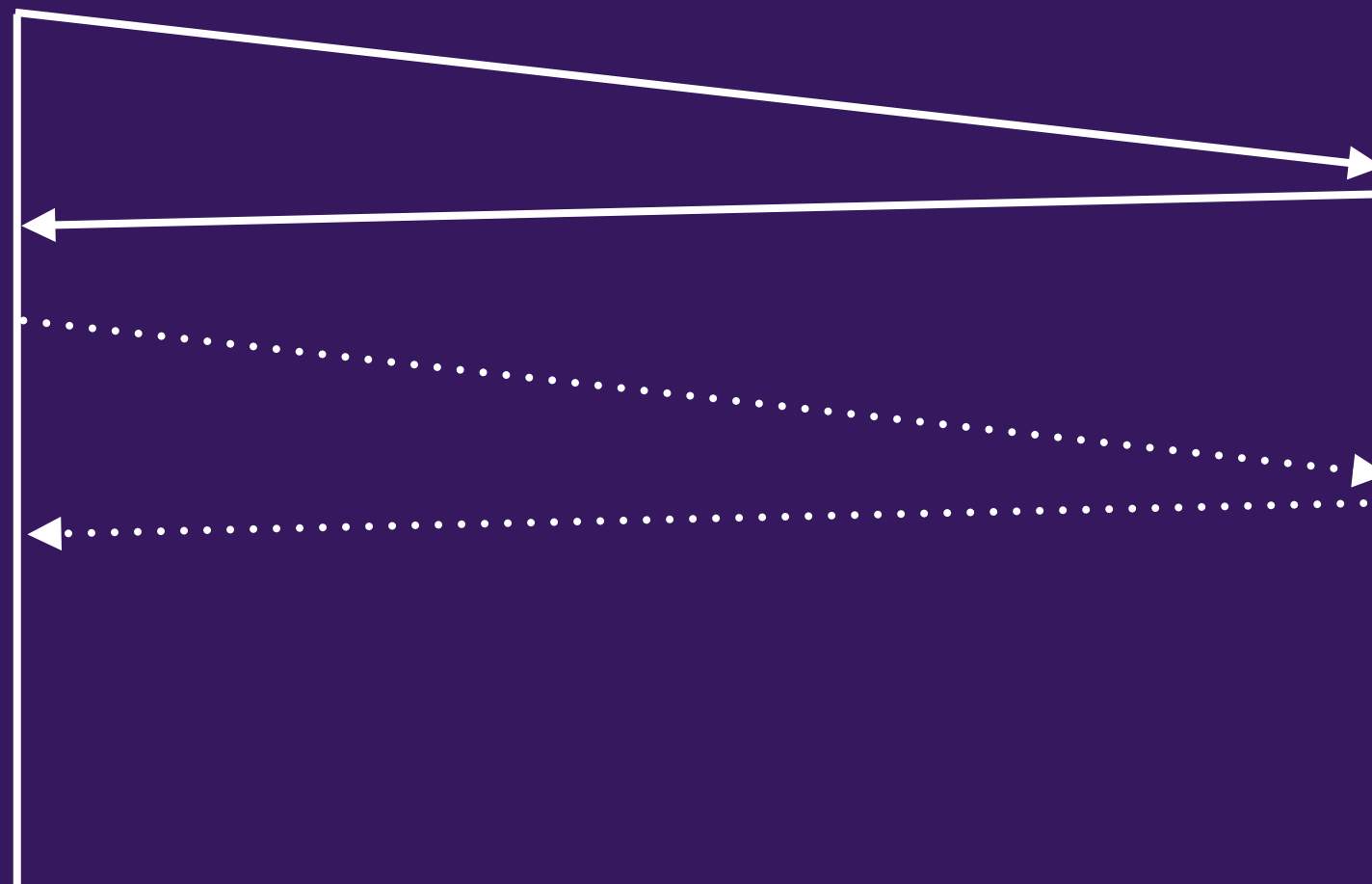
Fredrik Wärnsberg

# The Web in Real Time.

# The 90's The Request-Response Cycle



# 2005 Ajax



# 2006: Comet/Long Polling



Web 2.0:  
Now cleaner  
than ever



# 2006: Comet/Long Polling





# 2009: WebSockets



# How a Connection Is Established



# Browser Support & Fallback Strategies



10



**...use an existing library!**

# So, What Are My Options?

Java      jWebSocket, ...

.NET      SignalR, ...

JavaScript      Socket.IO, ...

Erlang      Cowboy, ...

Python      Tornado, ...

...

...

# WiFi Credentials

**Network: swisscom**  
**Password: Bmxx-3TG2**

# Demo Time!

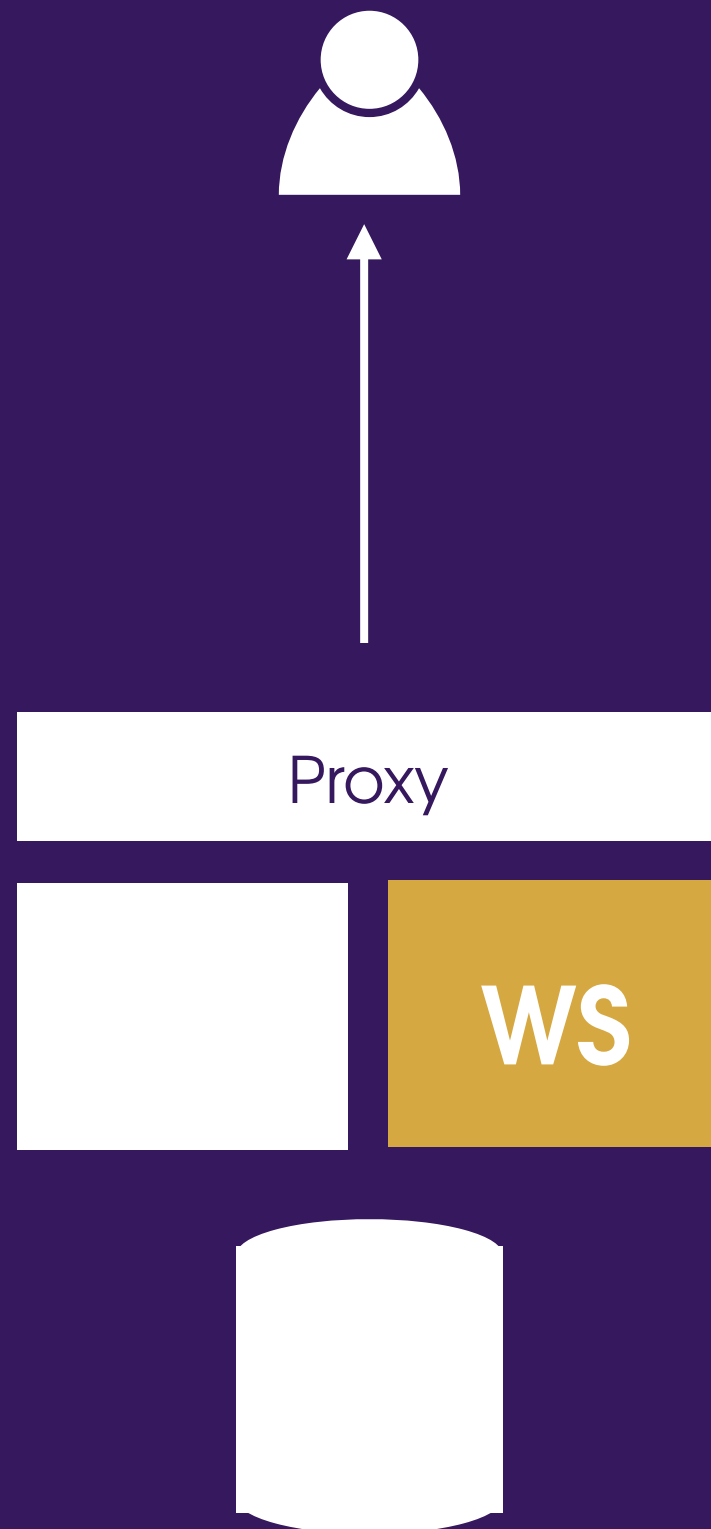
# Use Case #1: Monitoring

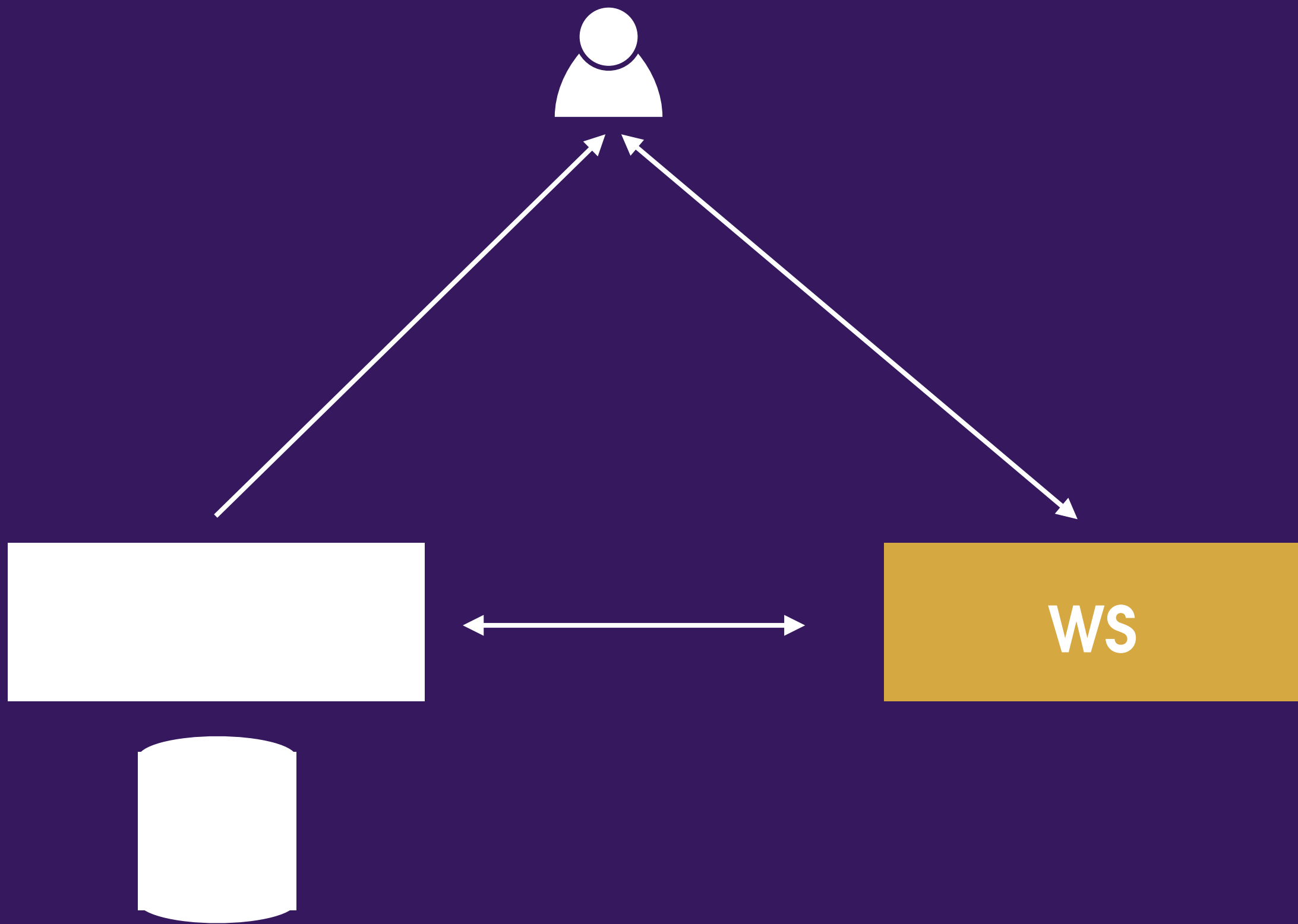
<http://fnnl.se>

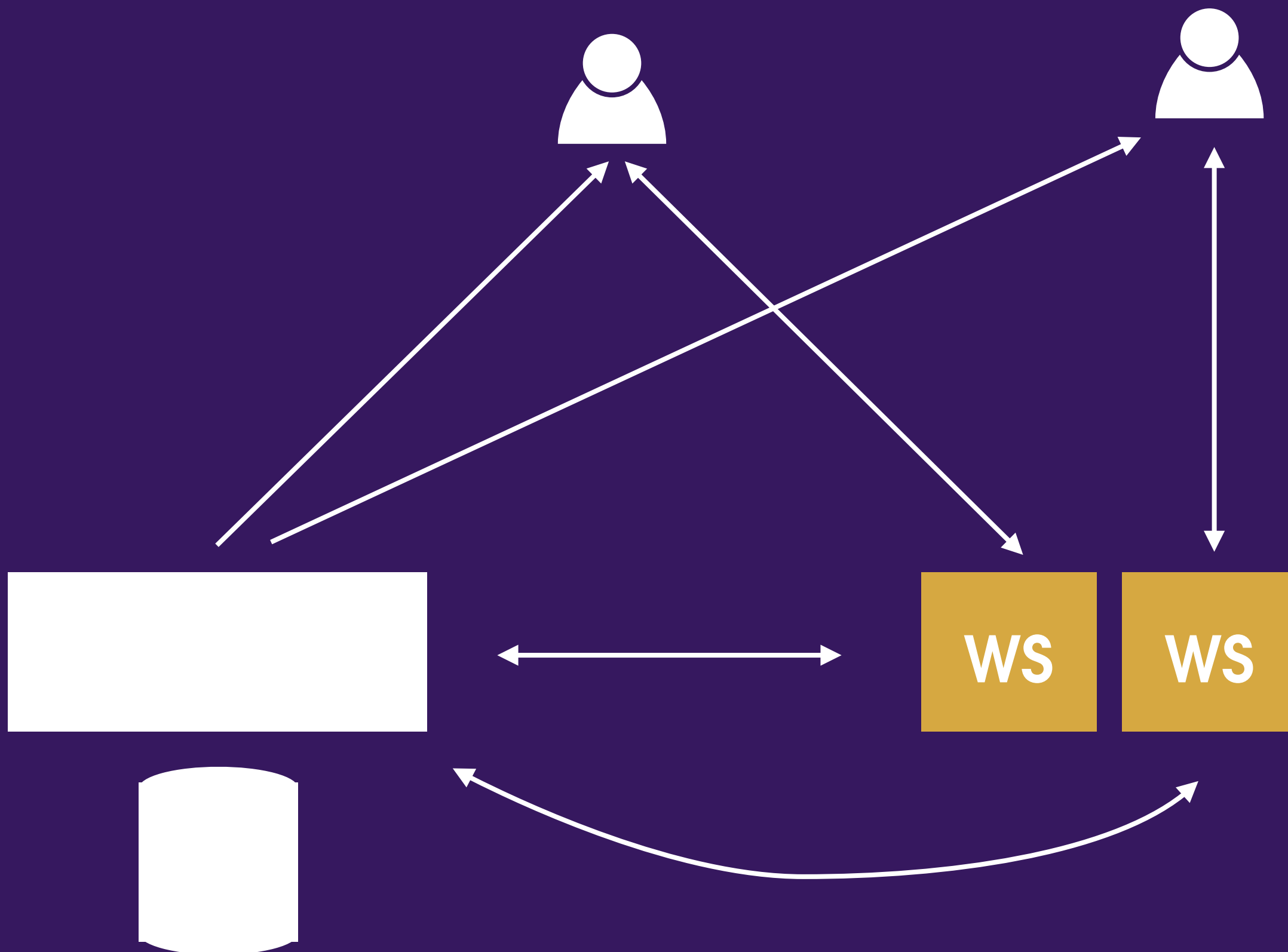












# Use Case #2: Collaboration

<http://fnnl.se>

```
this.socket.on('game:turn:next', this.didFinishTurn)
this.socket.on('game:end', this.didFinishRound)
this.socket.on('game:reset', this.didStartRound)
this.socket.on('game:player:joined', this.playerJoined)
this.socket.on('game:player:left', this.playerLeft)
```

# Smarter Clients



# Use Case #3: Games

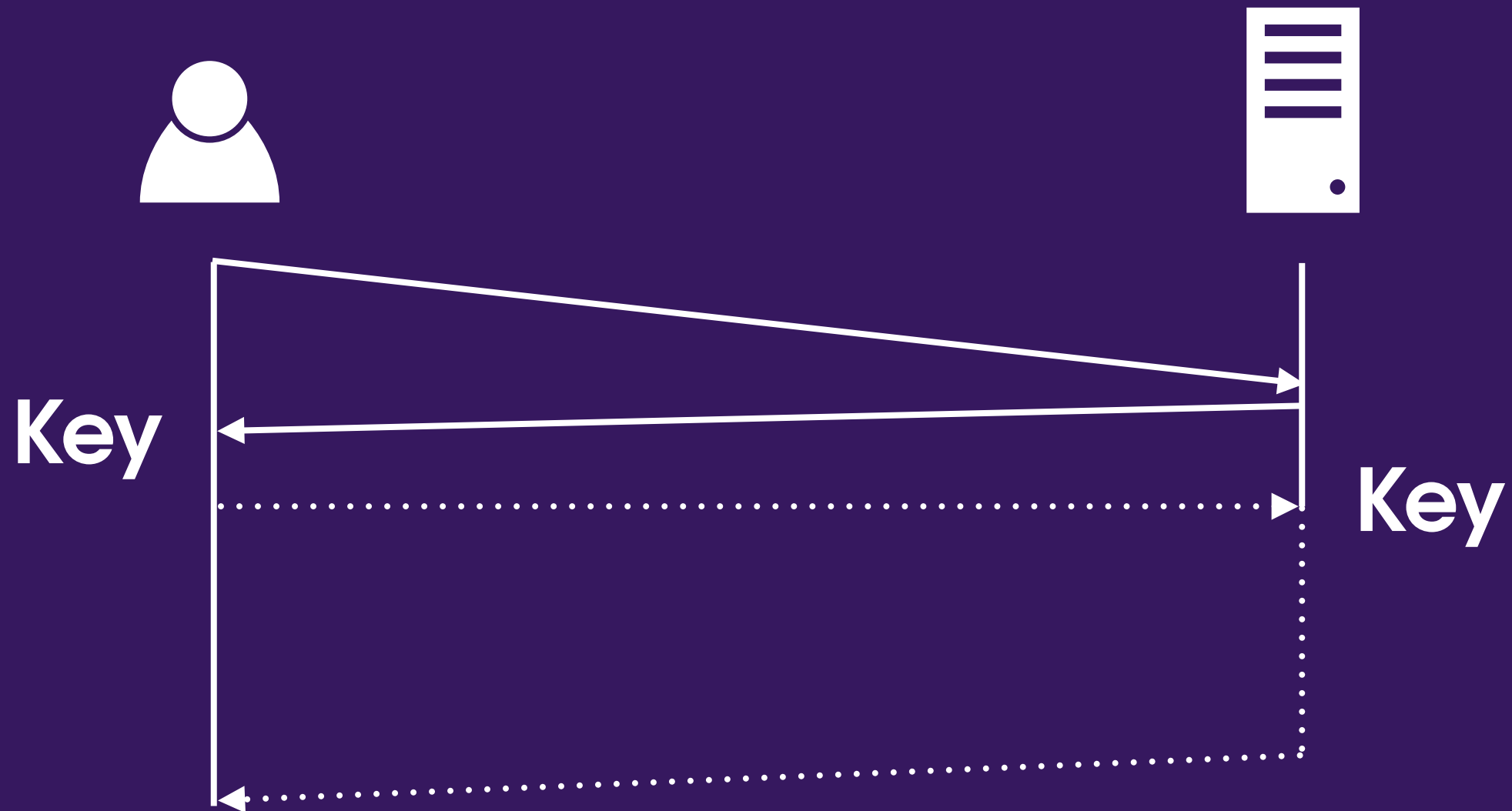
<http://fnnl.se>

# Defining Your Application Protocol

# The Future of WebSockets

<http://github.com/frekw/edge>

# Shared Sessions



```
game.on('objects collided', function(o1, o2)
{
    io.sockets.emit('objects collided',
        o1.id,
        o1.getState(),
        o2.id,
        o2.getState())
})
```

```
player.on('still', function() {  
    // Broadcast player still  
    io.sockets.emit('still player'  
        player.id,  
        player.getState()  
    })  
})
```

```
motion.on('move', function(direction) {  
    socket.emit('move', direction);  
})
```



```
socket.on('move', socket_move =  
function(direction) {  
    if (player == null) return;  
  
    // Start moving  
    player.move(direction)  
})
```