### 1、旧方案存在的技术痛点

爱奇艺号是我们内容生态的重要组成，作为前台系统，对用户体验有较高要求，直接影响着创作者的创作热情目前，爱奇艺号多个业务场景中用到了WebSocket实时推送技术，包括：

1）用户评论：实时的将评论消息推送到浏览器；

2）实名认证：合同签署前需要对用户进行实名认证，用户扫描二维码后进入第三方的认证页面，认证完成后异步通知浏览器认证的状态；

3）活体识别：类似实名认证，当活体识别完成后，异步将结果通知浏览器。

在实际的业务开发中我们发现，WebSocket实时推送技术在使用中存在一些问题。这些问题是：

1）首先：WebSocket技术栈不统一，既有基于Netty实现的，也有基于Web容器实现的，给开发和维护带来困难；

2）其次：WebSocket实现分散在在各个工程中，与业务系统强耦合，如果有其他业务需要集成WebSocket，面临着重复开发的窘境，浪费成本、效率低下；

4）最后：缺乏监控与报警，虽然可以通过Linux的Socket连接数大致评估WebSocket长连接数，但数字并不准确，也无法得知用户数等具有业务含义的指标数据；无法与现有的微服务监控整合，实现统一监控和报警。

2、新方案的技术目标

如上节所示，为了解决旧方案中存在的问题，我们需要实现统一的WebSocket长连接实时推送网关。这套新的网关需要具备如下特点：

1）集中实现长连接管理和推送能力：统一技术栈，将长连接作为基础能力沉淀，便于功能迭代和升级维护；

2）与业务解耦：将业务逻辑与长连接通信分离，使业务系统不再关心通信细节，也避免了重复开发，浪费研发成本；

3）使用简单：提供HTTP推送通道，方便各种开发语言的接入。业务系统只需要简单的调用，就可以实现数据推送，提升研发效率；

5）多端消息同步：允许用户使用多个浏览器或标签页同时登陆在线，保证消息同步发送；

6）多维度监控与报警：自定义监控指标与现有微服务监控系统打通，出现问题时可及时报警，保证服务的稳定性。

3、新方案的技术选型

在众多的WebSocket实现中，从性能、扩展性、社区支持等方面考虑，最终选择了Netty。Netty是一个高性能、事件驱动、异步非阻塞的网络通信框架，在许多知名的开源软件中被广泛使用。

Netty是为了快速开发可维护的高性能、高可扩展、网络服务器和客户端程序而提供的异步事件驱动基础框架和工具。换句话说，Netty是一个Java NIO客户端/服务器框架。基于Netty，可以快速轻松地开发网络服务器和客户端的应用程序。与直接使用Java NIO相比，Netty给大家造出了一个非常优美的轮子，它可以大大简化了网络编程流程。例如，Netty极大地简化TCP、UDP套接字、HTTP Web服务程序的开发。

Netty的目标之一，是要使开发可以做到“快速和轻松”。除了做到“快速和轻松”的开发TCP/UDP等自定义协议的通信程序之外，Netty经过精心设计，还可以做到“快速和轻松”地开发应用层协议的程序，如FTP, SMTP, HTTP以及其他的传统应用层协议。

Netty的目标之二，是要做到高性能、高可扩展性。基于Java的NIO, Netty设计了一套优秀的Reactor反应器模式。后面会详细介绍Netty中反应器模式的实现。在基于Netty的反应器模式实现中的Channel（通道）、Handler（处理器）等基类，能快速扩展以覆盖不同协议、完成不同业务处理的大量应用类。