

ĐIỆN TOÁN Đám Mây DI ĐỘNG – QUICKBLOX

Trần Thị Thúy*, Trần Lam Quyền**

TÓM TẮT

Trên thị trường di động ngày nay, có thể coi ứng dụng là một phần gắn liền với sự sống còn của một nền tảng di động. Android đã rất thành công với số lượng ứng dụng phong phú, đáp ứng đủ mọi nhu cầu của người dùng. Một phần nhỏ ứng dụng đã bắt đầu áp dụng công nghệ điện toán đám mây vì nhiều lý do và mức độ tiện ích khác nhau. Một trong những dịch vụ đám mây di động hiện nay được sử dụng phổ biến là Quickblox. Trong bài này, sẽ trình bày tổng quan về Quickblox và một số ứng dụng mẫu của Quickblox.

Từ khóa: điện toán đám mây, dịch vụ điện toán đám mây quickblox

ABSTRACT

In today's mobile market, applications can be considered to be integral to the survival of a mobile platform. Android has been very successful with a number of rich applications that meet all user needs. A small portion of the application has begun to apply cloud computing for a variety of reasons and levels of utility. One of the most popular cloud-based services is Quickblox. In this article, we'll give you an overview of Quickblox and some sample applications from Quickblox.

Keywords: cloud computing, quickblox

1. Điện toán đám mây (cloud computing)

1.1. Tổng quan

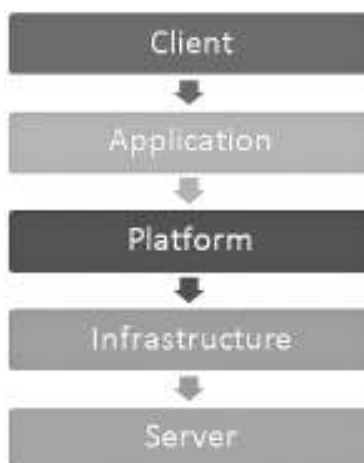
Theo định nghĩa của NIST (National Institute of Standards and Technology) [2], điện toán đám mây là một mô hình cho phép thuận tiện, truy cập mạng theo yêu cầu đến một nơi chứa các nguồn tài nguyên tính toán có thể chia sẻ và cấu hình được (ví dụ: mạng, máy chủ, lưu trữ, ứng dụng và dịch vụ), ở đó chúng có thể được cung cấp và phát hành nhanh chóng với nỗ lực quản lý hoặc tương tác với nhà cung cấp tối thiểu. Điện toán đám mây đôi khi còn được coi là thế hệ internet mới.

1.2. Cấu trúc

Theo tác giả [3], điện toán đám mây được chia làm 5 lớp như hình 1:

* Thạc sĩ, Khoa Kỹ thuật Công nghệ, trường Đại học Cửu Long

** Thạc sĩ, Phòng QLKH-SĐH&HTQT, trường Đại học Cửu Long



Hình 1. Cấu trúc điện toán đám mây

Client (lớp khách hàng): bao gồm phần cứng và phần mềm, khách hàng có thể truy cập và sử dụng các dịch vụ được cung cấp từ điện toán đám mây. Như máy tính và dây mạng kết nối internet (thiết bị phần cứng), các trình duyệt web (phần mềm),...

Application (lớp ứng dụng): làm nhiệm vụ phân phối phần mềm như một dịch vụ thông qua internet, người dùng không cần cài đặt và chạy các ứng dụng trên máy tính của mình, các ứng dụng dễ dàng được chỉnh sửa và người dùng sẽ nhận được hỗ trợ để biết được sự chỉnh sửa đó.

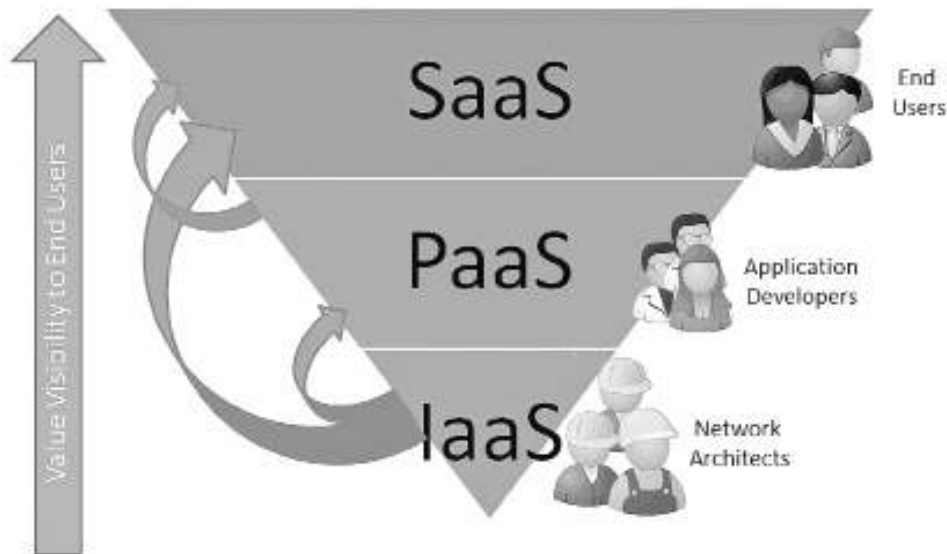
Platform (lớp nền tảng): cung cấp các nền tảng và giải pháp cho ứng dụng, chi phối đến cơ sở hạ tầng của đám mây và là điểm tựa cho lớp ứng dụng, cho phép các ứng dụng hoạt động trên nền tảng đó. Nó giảm nhẹ chi phí cho người dùng, vì không cần trang bị cơ sở hạ tầng (phần cứng và phần mềm).

Infrastructure (lớp cơ sở hạ tầng): cung cấp hạ tầng máy tính, tiêu biểu là môi trường nền ảo hoá. Thay vì, khách hàng phải bỏ tiền ra mua các server, phần mềm, trung tâm dữ liệu hay thiết bị kết nối,... thì giờ đây họ vẫn có thể sử dụng đầy đủ các tài nguyên mà chi phí giảm, thậm chí là miễn phí. Đây là một bước tiến bộ đáng kể của mô hình máy chủ ảo (Virtual Private Server).

Server (máy chủ): gồm các sản phẩm phần cứng và phần mềm máy tính, được thiết kế đặc biệt để cung cấp các dịch vụ của đám mây. Các máy chủ phải được xây dựng và cấu hình đủ mạnh (thậm chí là rất mạnh) để đáp ứng số lượng lớn người dùng và các nhu cầu ngày càng cao của họ

1.3. Mô hình về các dịch vụ

Theo [1], mô hình các dịch vụ của điện toán đám mây như hình 2:



Hình 2. Mô hình tổng quan dịch vụ của điện toán đám mây

IaaS (Infrastructure as a Service): nghĩa là người dùng có thể truy cập vào kiến trúc phần cứng mạng máy tính. Cung cấp nhiều nguồn tài nguyên như firewalls, load balancers, các địa chỉ IP. Nhưng hệ điều hành và các ứng dụng sẽ do người dùng tự cài đặt và cập nhật, điều này giúp người dùng linh động hơn trong việc sử dụng tài nguyên vào mục đích gì.

PaaS (Platform as a Service): hỗ trợ người dùng bằng các hệ điều hành, cơ sở dữ liệu, máy chủ web và môi trường thực thi lập trình, cho phép người dùng tập trung vào một ứng dụng cụ thể hơn là chi phí thời gian cho hệ điều hành, cho phép các nhà quản lý đám mây quản lý và đo đạc tài nguyên một cách tự động. Các nhà cung cấp IaaS cũng cung cấp PaaS giúp người dùng giảm tải công việc.

SaaS (Software as a Service): là sự phù hợp nhất khi tập trung vào người dùng cuối, giúp truy cập tới các phần mềm trên nền tảng đám mây mà không cần quản lý cơ sở hạ tầng và nền tảng nó đang chạy. Giúp người dùng dễ dàng truy cập và có khả năng mở rộng. SaaS gồm email, phần mềm văn phòng, các công cụ kiểm toán từ Google, Microsoft, Freshbooks,...

2. Công nghệ điện toán đám mây Quickblox

2.1. Giới thiệu chung

Quickblox là một đám mây tổ chức nền tảng (PaaS) được thiết kế để đơn giản hóa việc phát triển ứng dụng di động [5].

Quickblox là khối xây dựng được gắn vào các ứng dụng trên thiết bị di động để thêm chức năng mới, làm giảm thời gian và chi phí phát triển, khả năng mở rộng và ứng dụng cao, giảm thiểu khả năng đồng bộ hóa các ứng dụng mới và hiện có với các chức năng xã hội và tương tác.

Quickblox là nền tảng điện toán đám mây được lưu trữ trên máy chủ cung cấp các lợi ích với các phần mềm truyền thống lưu trữ và cơ sở hạ tầng:

- Tốc độ thời gian đưa ra thị trường nhanh hơn – Đăng kí trên web để có được các ứng dụng với Quickblox.
- Ít nhu cầu hỗ trợ công nghệ thông tin: Quickblox's SLA cung cấp cho khách hàng doanh nghiệp với định mức hỗ trợ và một bản trợ giúp chuyên dụng.
- Nâng cấp dễ dàng hơn: các tính năng Quickblox mới nhất và các chức năng có sẵn, tùy chỉnh và cấu hình thông qua bảng quản trị của riêng.

Phụ trợ ứng dụng được lưu trữ với Amazon (EC2) với một trường hợp chia sẻ chuyên dụng cùng với mật mã và xác thực – vì vậy dữ liệu được an toàn. Không giống như BlackBox giải pháp từ PaaS khác, quyền kiểm soát máy chủ là của người đăng kí tài khoản Quickblox và tất nhiên có quyền truy cập vào mã nguồn của hệ thống Quickblox.

Quickblox mang lại giá trị cho ứng dụng đảm bảo các ứng dụng của bản kế tiếp phát triển với thời gian ngắn hơn.

2.2. Quickblox User

- Tài liệu, hồ sơ kết hợp và tích hợp với bên thứ 3 (facebook, twitter) cho phép người dùng chia sẻ rộng rãi ứng dụng.
- Tính năng xác thực.
- Tính năng nhận biết.
- Tương thích với facebook.
- Bổ sung dữ liệu trên các lĩnh vực hỗ trợ tùy chỉnh (hình ảnh, hình đại diện, thông tin của người dùng thông tin).
- ID là duy nhất và khai thác dữ liệu trên các module.

2.3. Quickblox Custom Object

- Tính năng tạo ra các cơ sở dữ liệu phù hợp với nhu cầu của người sử dụng
- Tính năng truy cập và gửi dữ liệu nhanh chóng.
- Tùy chỉnh đối tượng cung cấp sự linh hoạt để xác định cấu trúc của dữ liệu mà người dùng cần.
- Khả năng thêm mới, chỉnh sửa, xóa đơn giản.

2.4. Các phiên bản và tính năng của QuickBlox SDK

3.2.0 – 22/11/2016

- Chức năng cốt lõi:
 - Quản lý tự động. Bắt đầu từ QuickBlox Android SDK 3.2 không cần phải quản lý thủ công. SDK sẽ tự động làm tất cả các công việc này.
 - Loại bỏ các API lỗi thời từ các phiên bản 2.x
- Chức năng Push:

- Đăng kí tự động, không còn lo lắng làm thế nào để có được GCM hay FCM, đăng kí hay làm gì với dữ liệu nhận được trong các đoạn mã.

- Chức năng trò chuyện:

- Bổ sung chức năng gửi các thông báo không đồng bộ

- Chức năng trò chuyện mở rộng:

- Bổ sung mô đun trò chuyện mở rộng với BOSH, cho phép ứng dụng thiết lập XMPP trên Proxy

- Chức năng nội dung:

- Cố định SSLHandshakeException trong khi tải các tập tin lên các thiết bị có hệ điều hành trước Android 4.3

- Kích thước mô đun giảm rất nhiều lần

3.1.0 - 26/10/2016

- Chức năng trò chuyện:

- Cải thiện cơ chế đăng nhập, thời gian nhanh hơn 2 lần so với trước

- Bỏ qua các trạng thái tin nhắn trong QBIncomingMessagesManager

- Cố định QBChatDialog khi tải từ máy chủ về mà không cần đang kết nối trò chuyện

- Cố định trò chuyện 3.0 bugs

- Chức năng mở rộng:

- Xóa bỏ các phụ thuộc không cần thiết với các mô đun mở rộng

3.0 - 16/9/2016

- Chức năng trò chuyện

- Đổi tên **QBDialog** thành **QBChatDialog** cho iOS

- **QBChatDialog** chịu trách nhiệm cho tất cả các hoạt động liên quan đến trò chuyện – gửi/nhận tin nhắn, tham gia, trạng thái gõ,...

- Thêm **listeners** cho **QBChatDialog**

- Lớp **GroupChat** và **PrivateChat** không được tán thành

- Thêm **QBIncomingMessagesManager** để nghe tin nhắn từ tất cả hội thoại

- Thiết lập **QBChatService**: khả năng thiết lập cổng, khả năng lắng nghe những thay đổi mạng để kết nối lại nhanh, khả năng tham gia tự động khi tải hoặc tạo **QBChatDialogs**

- Danh sách riêng tư: thêm khả năng để người dùng chặn lẫn nhau

- Tất cả listeners được gọi bây giờ được quản lý bởi tiến trình UI

- Cải thiện hiệu suất khi tham gia nhiều hội thoại cùng một lúc

- Chức năng API:

- Thêm giao diện **Performer** để đại diện cho bất kì phương thức REST API nào. Performer có thể được thực hiện đồng bộ và hủy bỏ bất cứ lúc nào. Tất cả các phương thức REST API từ phiên bản SDK 2.x trở đi có hậu tố V2 và hủy bỏ.

- Chức năng mở rộng
- Thêm mô đun mở rộng cho các chức năng SDK thử nghiệm và mở rộng
- Thêm hỗ trợ RxJava và chuyển đổi giao diện SDK Performer thành rx Observable

2.6.1 - 22/7/2016

- Chức năng gọi WebRTC:
 - Cải thiện WebRTC để thực hiện gọi và giao diện đơn giản
 - Thêm tất cả cuộc gọi lại vào tiến trình UI
 - Thêm tính năng cài đặt fps, thay đổi độ phân giải trong suốt quá trình gọi, chuyển đổi giữa các phần cứng/phần mềm AEC, khả năng xử lý vô hiệu hóa âm thanh AEC, giảm tiếng ồn,...khả năng kích hoạt/vô hiệu hóa sử dụng OpenSL ES để phát lại âm thanh
 - Thêm **QBRTCSurfaceView** để hiển thị khung hình cuộc gọi và nhiều hình chồng lên nhau trên một màn hình chính
 - Sửa một số lỗi nhỏ
- Chức năng trò chuyện:
 - Thêm khả năng cho phép/vô hiệu hóa cơ chế TLS trong trò chuyện qua **QBChatService**.

ConfigurationBuilder

- Thêm khả năng thiết lập tùy chỉnh bối cảnh SSL thông qua **QBChatService**.

ConfigurationBuilder

- Chức năng nội dung:
 - Sửa lỗi xử lý ngoại lệ **SignatureDoesNotMatch** khi tập tin lên
 - Áp dụng phương pháp tải tập tin lên để làm việc với kho tập tin khác
- Khác: Xây dựng lại mô đun SDK và thiết lập các SDK phụ thuộc đến project thông qua gradle

2.6 - 14/6/2016

- Chức năng gọi WebRTC:
 - Thêm khả năng lấy dữ liệu kết nối ngang hàng trong suốt cuộc gọi. Thêm **QBRTCStatsReportCallback** vào lớp **QBRTCSession**
- Chức năng trò chuyện:
 - Cập nhật phương thức equals và hashCode trong mô hình **QBDialog** để có thể lưu trữ và tìm kiếm trong Bộ sưu tập
 - Thêm **QBMessageUpdateBuilder** để đơn giản hóa việc cập nhật tin nhắn
 - Thêm **QBDialogRequestBuilder** để đơn giản hóa việc cập nhật hộp thoại
 - Thêm **QBMessageGetBuilder** để đơn giản hóa việc yêu cầu tin nhắn của các bộ lọc
 - Thêm yêu cầu API trò chuyện: bắt cuộc có thông báo khi xóa hộp thoại, xóa tin nhắn, xóa nhiều hộp thoại trong truy vấn, cập nhật tin nhắn trò chuyện
 - Thêm trả về joinFailed khi người dùng cố gắng tham gia phòng không tồn tại

- Thêm khả năng quản lý cài đặt trò chuyện – đang trò chuyện, thời gian chờ, thời gian mặc định. Thêm lớp **ConfigurationBuilder** vào lớp **QBChatService**

- Sửa lỗi carbon cho hệ thống tin nhắn

- Chức năng tùy chỉnh đối tượng:

- Thêm yêu cầu cho chức năng tập hợp lại: **QBCustomObjects.getAggregationObjects()**.

- Chức năng cho mô đun nội dung:

- Sửa phương thức gọi **onProgressUpdate** cho **QBProgressCallback** cho tiến trình upload/download được bắt đầu. Nếu yêu cầu bắt đầu từ **UIThread** Android, **QBProgressCallback** sẽ được gọi **UIThread** mặc dù tiến trình này được chạy trong tiến trình nền

- Sửa **QBProgressCallback** đúng tiến trình trong các yêu cầu tải

- Mô đun lỗi: Sửa lỗi **NPE** trong khi phân tích yêu cầu url với các thay đổi của bộ lọc

2.5.2 - 15/4/2016

- Chức năng gọi video:

- Cải thiện **WebRTC** để gọi ổn định hơn

- Sửa một số lỗi chính và dụng độ

- Phương thức **hangUp** và **reject** (gác máy và từ chối) cuộc gọi gán tham số **userInfo** thích hợp đến **onReceiveHangUpFromUser** và **onCallRejectByUser**

- **QBRTCSession** không quản lý các cài đặt về âm thanh nữa. Thêm lớp **AppRTCAudioManager** để quản lý âm thanh thủ công.

- Chỉnh sửa cuộc gọi video. Nếu thực hiện cuộc gọi nhưng thiết bị không có máy ảnh hay chưa cho quyền truy cập máy ảnh, thiết bị sẽ không gửi video đến bên kia nhưng có thể nhận video từ bên kia.

- Chức năng trò chuyện:

- Thêm tính năng tự động gửi khi có kết nối mạng, xử lý trong phương thức **startAutoSendPresence** và **stopAutoSendPresence** trong **QBChatService**

- Sửa lỗi về thứ tự sai khi nhận tin nhắn

- Thêm truy vấn để lấy tổng số tin nhắn chưa đọc cho hộp thoại **QBChatService**. **getTotalUnreadMessagesCount**

- Thêm trường **updated_at** thành **'toString'** trong lớp **QBDialog**

- Thêm **NPE** trong phương thức kết nối không đồng bộ với callback là null trong trò chuyện nhóm

- Sửa lỗi **OutOfBoundsException** khi lấy dữ liệu vị trí từ **QBDialog**

- Sửa lỗi thiết lập tài nguyên khi trò chuyện theo Id cho thiết bị là duy nhất. Điều này hữu ích khi đăng nhập một tài khoản trên nhiều thiết bị khác nhau

- Khác: Sửa lỗi truy vấn **QBUsers.updateUser**, để xóa tag trống từ **StringifyArrayList** thành phương thức **QBUser.setTags**, nếu đặt null thì tag không thể thay đổi được

2.5.1 - 3/3/2016

– Sửa lỗi NullPointerException ở sự kiện “accept” trong mô đun trò chuyện video của lớp QBRTCSession trong một số trường hợp

2.5 - 5/2/2016

- Phục vụ thiết bị đầu cuối tự động. Đây là vấn đề chuyển đổi mượt giữa các dự án
- Cập nhật cách SDK khởi tạo

QBSettings.getInstance().init(getApplicationContext(), APP_ID, AUTH_KEY, AUTH_SECRET);

QBSettings.getInstance().setAccountKey(ACCOUNT_KEY);

- Cập nhật cách SDK tùy chỉnh trở đến máy chủ

QBSettings.getInstance().setEndpoints(API_DOMAIN, CHAT_DOMAIN), ServiceZone.PRODUCTION);

QBSettings.getInstance().setZone(ServiceZone.PRODUCTION);

– Đơn giản hóa việc khởi động trò chuyện, phương thức **QBChatService.isInitialized()** và **QBChatService.init(context)** có hiệu lực ngay phiên bản này

- Cập nhật lỗi callback:

QBEntityCallback.onError(List<String> errors) -> QBEntityCallback.onError(QBResponseException exception)

- Tất cả thư viện bên thứ ba được tích hợp vào SDK

– **Chat Stream Management** (XEP-0198), trạng thái “gửi” trong trò chuyện, thêm giao diện **QBMessageSentListener**

- Loại bỏ các phương thức lỗi thời của API 1.x. xóa lớp Result

- Đổi tên QBMessages thành **QBPushNotifications**

- Đơn giản hóa việc mô tả API Push

– Không dùng **QBPushNotifications.subscribeToPushNotificationsTask(registrationID, deviceId, QBEnvironment.DEVELOPMENT)** và thay thế bằng **QBPushNotifications.createSubscription(subscription)**

- Nội dung: truy cập url public/private

• **QBFile.publicUrl, QBFile.publicUrlForID, QBFile.privateUrl, QBFile.publicUrlForID**

- Sửa lỗi **QBRequestUpdateBuilder** ‘pullAll’ và thêm toán tử ‘pushAll’

- Đổi tên tất cả setters và getters thành ý nghĩa đúng trong **QBLimitedRequestBuilder**

- Dọn dẹp **QBEntityCallback** và xóa **QBEntityCallbackImpl**

- Chuyển sang thư viện ‘**HttpURLConnection**’ thay vì Apache Client

- Thêm phương thức **QBChatService.getChatDialogsCount()** để lấy số lượng hộp thoại

- Bỏ **TransferProtocol**
- Tích hợp các chữ số Twitter
- Sửa trường **QBChatMessage.dateSent** dùng làm nền tảng
- Thêm khả năng gửi Push cụ thể của Apple
- **QBChatMessage**: đặt **senderID** để đọc và cấp phát các Id
- Truy cập **total_entries** trong yêu cầu các hộp thoại trò chuyện
- Thêm lớp **QBDialogCustomData** để làm việc với dữ liệu hộp thoại trò chuyện tùy chỉnh
- Xử lý lỗi ngoại lệ **org.apache.http.conn.HttpHostConnectException** khi sử dụng **REST API** và khi không có kết nối mạng

- Xóa bỏ API địa điểm

2.4 - 27/11/2015

- Nhóm WebRTC gọi video
- Loại bỏ thư viện trò chuyện video cũ

2.3.3 - 5/11/2015

- Cập nhật url trong phương thức **QBFile.getPublicUrl**

2.3.2 - 26/10/2015

- Tính năng:
 - Hệ thống API thông báo
 - Khả năng thiết lập “**QBChatMessage.senderId**”, hữu ích để gửi trạng thái “read” trong trường hợp không biết nguồn gốc của đối tượng **QBChatMessage**
- Sửa lỗi:
 - Sửa lỗi các **NullPointerException** trong **QBJsonErrorParser**
 - Sửa lỗi khi cố xóa một đối tượng tùy chỉnh thông qua **QBCustomObjects.deleteObjects**
 - Sửa lỗi các vấn đề về mảng nhúng trong trường **CustomObjects**
 - Sửa lỗi về một số vấn đề về việc không gửi tin nhắn được giữa các **Android SDK** và **XMPP** máy tính bàn như **Adium** hay **Psi**
 - Sửa lỗi phương thức **QBRTCClient** ‘rejectCall’, nơi có tham số **userInfo** không bao giờ được sử dụng
 - Sửa lỗi với ngữ cảnh trong **QBRTCClient**
 - Nội dung:
 - Sửa lỗi vấn đề **OutOfMemory** khi tải tập tin có dung lượng lớn
 - Sửa lỗi vấn đề với **QBProgressCallback** khi nó tạo ra nhiều tiến trình tương tự
 - Cập nhật **QBFile.lastReadAccessTime** có kiểu Date

- Cập nhật phương thức đầu ra `QBFile.getPublicUrl` có định dạng mới
- Khác: `QBFile`: loại bỏ các thuộc tính không tác dụng như ‘`referencesCount`’ và ‘`lifeTime`’

2.3.1 - 14/9/2015

- Sửa lỗi vấn đề máy ảnh với cuộc gọi video WebRTC (r9806)
- `QBGroupChat` throws `IllegalStateException` nếu người không được kết nối trò chuyện
- Thêm getter **`QBChatMessage.isDelayed`** để phát hiện tin nhắn đã gửi khi người người dùng offline

2.3 - 6/9/2015

- Cuộc gọi video WebRTC
- Sửa lỗi **`NullPointerException`**
- Thiết lập giá trị mặc định **`QBRTCCConfig.dialingTimeInterval`** là 5 và **`QBRTCCConfig.answerTimeInterval`** là 60
- Cập nhật **`webrtc`** thành r9813
- Trò chuyện:
- Thêm “tối ưu hóa di động” – một cách để tiết kiệm pin khi trạng thái không hoạt động
- Thêm quản lý ping
- Thêm trường **`deliveredIds`** vào mô hình **`QBChatMessage`**
- Đánh giá các quản lý về trạng thái delivered/read, thêm lớp **`QBMessageStatusListener`** và **`QBMessageStatusesManager`**
- Thêm listener “đăng nhập...” cho nhóm trò chuyện
- Thêm **`getDialogId`** vào mô hình **`QBPrivateChat`** và **`QBGroupChat`**
- Nội dung: Sửa lỗi các vấn đề upload tập tin trên một vài thiết bị Android

2.2.6 - 13/7/2015

- WebRTC – đơn giản hóa vòng đời nền gọi, không cần thiết thực hiện các giao diện **`RTCSignallingMessageProcessorCallback`** để xử lý cuộc gọi có hình nền. Gỡ bỏ các truy cập public trong phiên bản SDK tiếp. Kể từ phiên bản này, không cần phải gọi phương thức **`prepareToProcessCalls(Context context)`** trong Activity, có thể gọi phương thức này cùng với phương thức thiết lập **`RTCClient`** khác trong bất kỳ thành phần Android nào

2.2.5 - 19/6/2015

- Cập nhật WebRTC SDK (r9417) và các bản sửa lỗi

2.2.4 - 7/6/2015

- Sửa lỗi tải tập tin với tiến trình callback

2.2.3 - 8/5/2015

- Sửa lỗi WebRTC, không thể thực hiện cuộc gọi sau khi đã kết thúc cuộc gọi trước

2.2.2 - 24/4/2015

- Sửa một số lỗi khi tải tập tin lên máy chủ QuickBlox khi kết nối mạng không ổn định
- Sửa lỗi về một số kí tự đặc biệt của tài khoản hay mật khẩu đăng nhập
- Phương thức QBChatService.getInstance().getPrivacyListsManager().setPrivacyList throws IllegalArgumentException trong QBPrivacyList là null hoặc QBPrivacyList.items là null hoặc **QBPrivacyList.items.size** là 0

2.2.1 - 3/4/2015

- Dãy trò chuyện nhóm
- Sửa lỗi NPE khi sử dụng cách cũ để tạo ra những phòng trò chuyện nhóm

2.2 - 30/3/2015

- Cập nhật WebRTC API cho cuộc gọi video

2.1.1 - 17/3/2015

- Thêm tất cả bộ lọc vào QBRequestGetBuilder
- QBUsers.getUsers trả về một hằng số. TOTAL_PAGES trong các kết quả trả về.
- Sửa một số lỗi ngoại lệ trong giao diện QBParticipantListener
- Sửa lỗi NullPointerException trong lớp QBGroupChat
- Thiết lập thêm khả năng cho trường -1 trong **QBUser** để xóa nó

3. Một số ứng dụng mẫu của QuickBlox SDK 3.2

Tiêu đề	API	Mô tả Hình ảnh minh họa
Simple Android Chat code sample	Chat, Users	– Chat ngang hàng – Chat nhóm – Chat dựa trên vị trí – Gửi thông báo cho người dùng offline 

Simple Android WebRTC (VideoChat) code sample	Chat, VideoChat	– Kích hoạt tính năng gọi video tương tự như FaceTime hay Skype 
Simple Android Location code sample	Location	– Sử dụng vị trí của GPS, tích hợp bản đồ và dữ liệu cục bộ – Người dùng có thể “nhìn thấy” nhau trên bản đồ và gắn kết với người dùng khác
Simple Android Push Notifications code sample	Messages, Users	– Cho phép gửi thông báo và cảnh báo cho người dùng ở bất cứ lúc nào thông qua một API – Có thể lọc các tin nhắn thông qua các tham số bổ sung 
Simple android Custom Objects (key value data store) code sample	Custom Objects	– Cho phép tạo ra bất kì cấu trúc dữ liệu, logic và một vài tùy chỉnh từ máy chủ 

Simple android Users authentication code sample	Users	– Giúp quản lý và duy trì người dùng trên bất kì nền tảng gì 
Simple android content storage and update code sample	Content	– Cho phép gửi, phân luồng, lưu trữ, chia sẻ, cập nhật và lưu nội dung tự động 

TÀI LIỆU THAM KHẢO

- [1] Hassan, Qusay, “Demystifying Cloud Computing” . *The Journal of Defense Software Engineering*. CrossTalk. 2011 (Jan/Feb): 16–21. Retrieved 11 December 2014
- [2] Peter Mell and Timothy Grance, *The NIST Definition of Cloud Computing*, National Institute of Standards and Technology: U.S. Department of Commerce.doi:10.6028/NIST.SP.800-145. Special publication 800-145.
- [3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, “Above the Clouds: A Berkeley View of Cloud Computing”. University of California, Berkeley, Feb 2009
- [4] Trần Cao Đệ, “Tổng quan về an ninh trên điện toán đám mây,” Tạp chí ĐHCT, số chuyên đề (2013), p. 39-46, ĐHCT, 2013.
- [5] http://quickblox.com/developers/QuickBlox_Developers

Ngày nhận bài: 05/1/2017

Ngày gửi phản biện: 21/1/2017