RICOH THETA x IoT
Developers Contest

Cloud API Seminar

THETA DEVELOPERS

UNOFFICIAL
COMMUNITY

Unofficial Translation of RICOH Presentation
Unauthorized

May 21, 2016

Agenda

1. Introduce RICOH Cloud API
2. Video Communication
3. Photo and Media Storage
4. Q&A
5. Future Plans

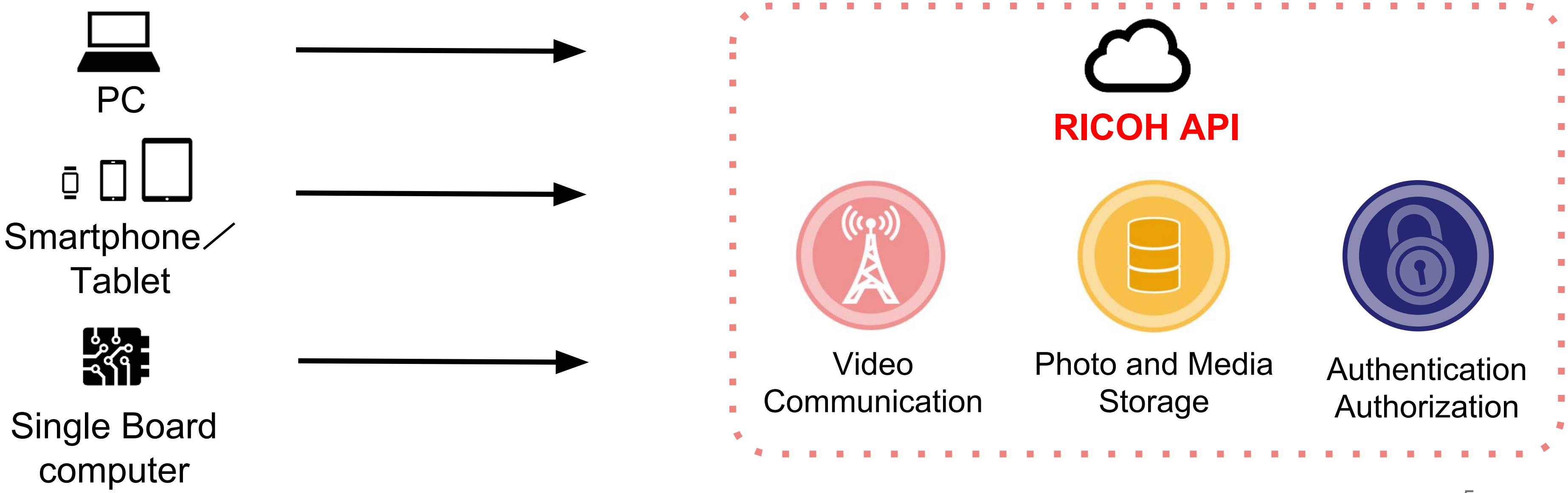
1. Introducing Cloud API

Unofficial translation of presentation originally created by
Junichi Takauwa
RICOH New Technology Development HQ
SV Technology Development Center, Platform Development Lab

Question

RICOH Cloud API

- Works with PC、Smartphone/Tablet、Single board computer (IoT)
- Suggested usage
  1. Video Communication
  2. Photo and Media Storage
  3. Authentication and Authorization
- Free beta version

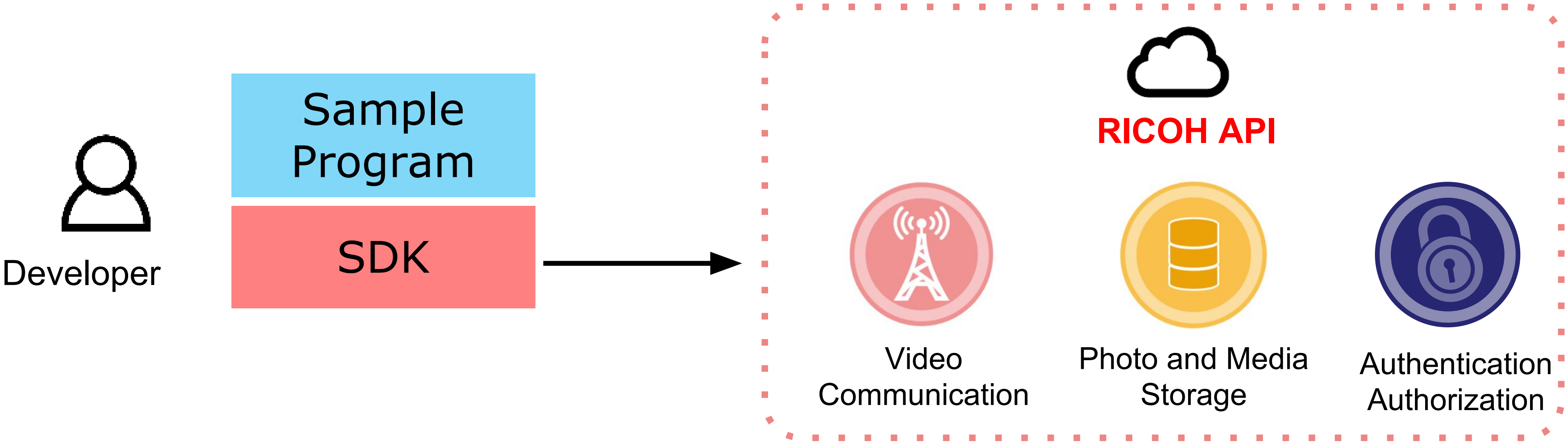


Resources for Developers

UNOFFICIAL GUIDES

- Available to Developers
  1. Cloud API
  2. SDK
  2. Sample Program

- On GitHub
  https://github.com/ricohapi/



Requirements

Needed to Use RICOH Cloud API

1. **Client Credentials** *To identify the app being used

・Client ID

・Client Secret

2. **User Account** *To identify the user

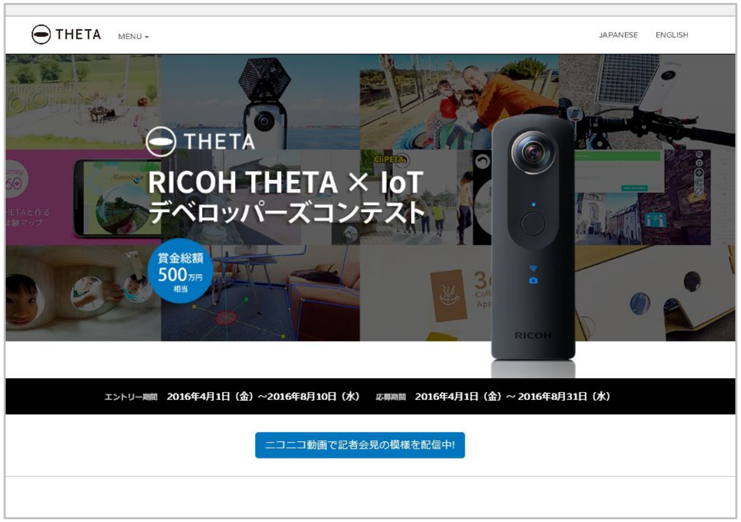
・User ID

・Password

Prepare in Advance
～Setting Client Credentials～

Step1. Fill out the requirements on the contest site and enter the contest!

http://contest.theta360.com/



RICOH THETA × IoT
デベロッパーズコンテスト

エントリー期間　2016年4月1日（金）～2016年8月10日（水）　応募期間　2016年4月1日（金）～2016年8月31日（水）

Step2. Follow the steps in the authentication email

Step3. Receive the contest registration confirmation email

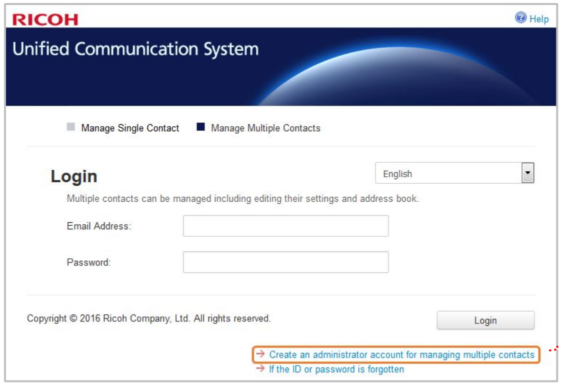
Step4. After that, following the instructions in the API explanation email, download your API credentials

Prepare in Advance ~Setting User Account~

Step 1) Go to the RICOH Unified Communication System administration page

https://beta2.ucs.ricoh.com/dashboard/login

Step 2) Click on the link



RICOH

Help

Unified Communication System

Manage Single Contact

Manage Multiple Contacts

Login

English

Multiple contacts can be managed including editing their settings and address book.

Email Address:

Password:

Copyright © 2016 Ricoh Company, Ltd. All rights reserved.

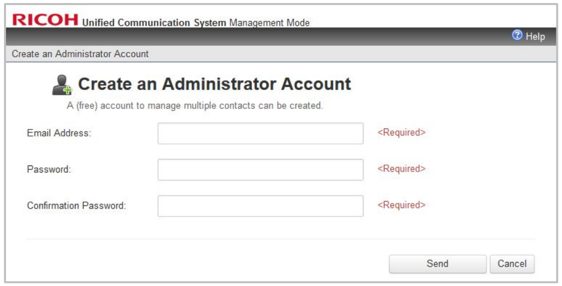
Login

→ Create an administrator account for managing multiple contacts

→ If the ID or password is forgotten

→ Create an administrator account for managing multiple contacts

Step 3) Fill out the form, click Send



RICOH Unified Communication System Management Mode

Help

Create an Administrator Account

Create an Administrator Account

A (free) account to manage multiple contacts can be created.

Email Address: <Required>

Password: <Required>

Confirmation Password: <Required>

Send Cancel

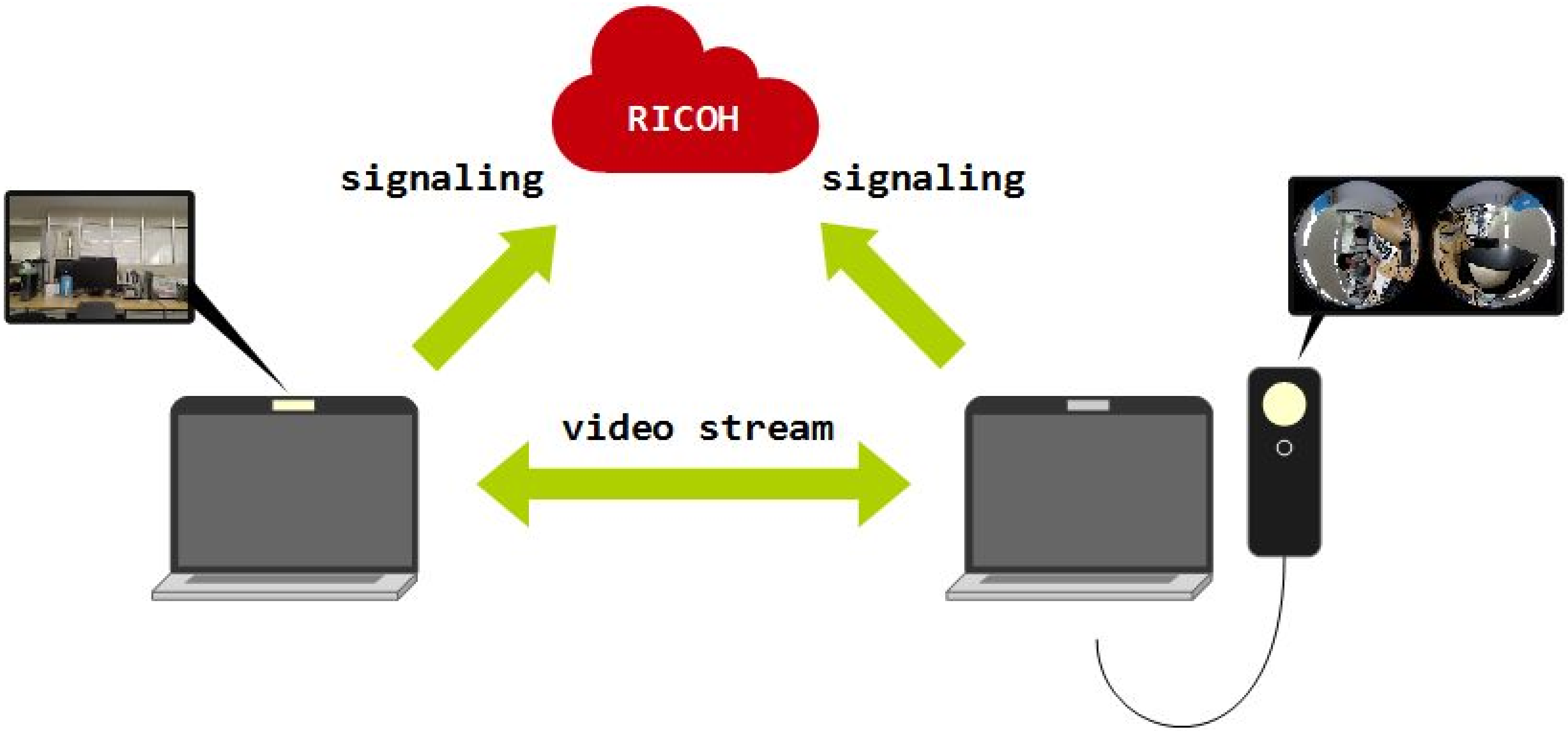
Step 4) Click on the URL in the authentication email. Your email address will be used as the user ID.

2. Video Communication

Originally presented by
Tomonori Aigawa
RICOH New technology Development HQ
SV Technology Development Center, Platform Development Lab

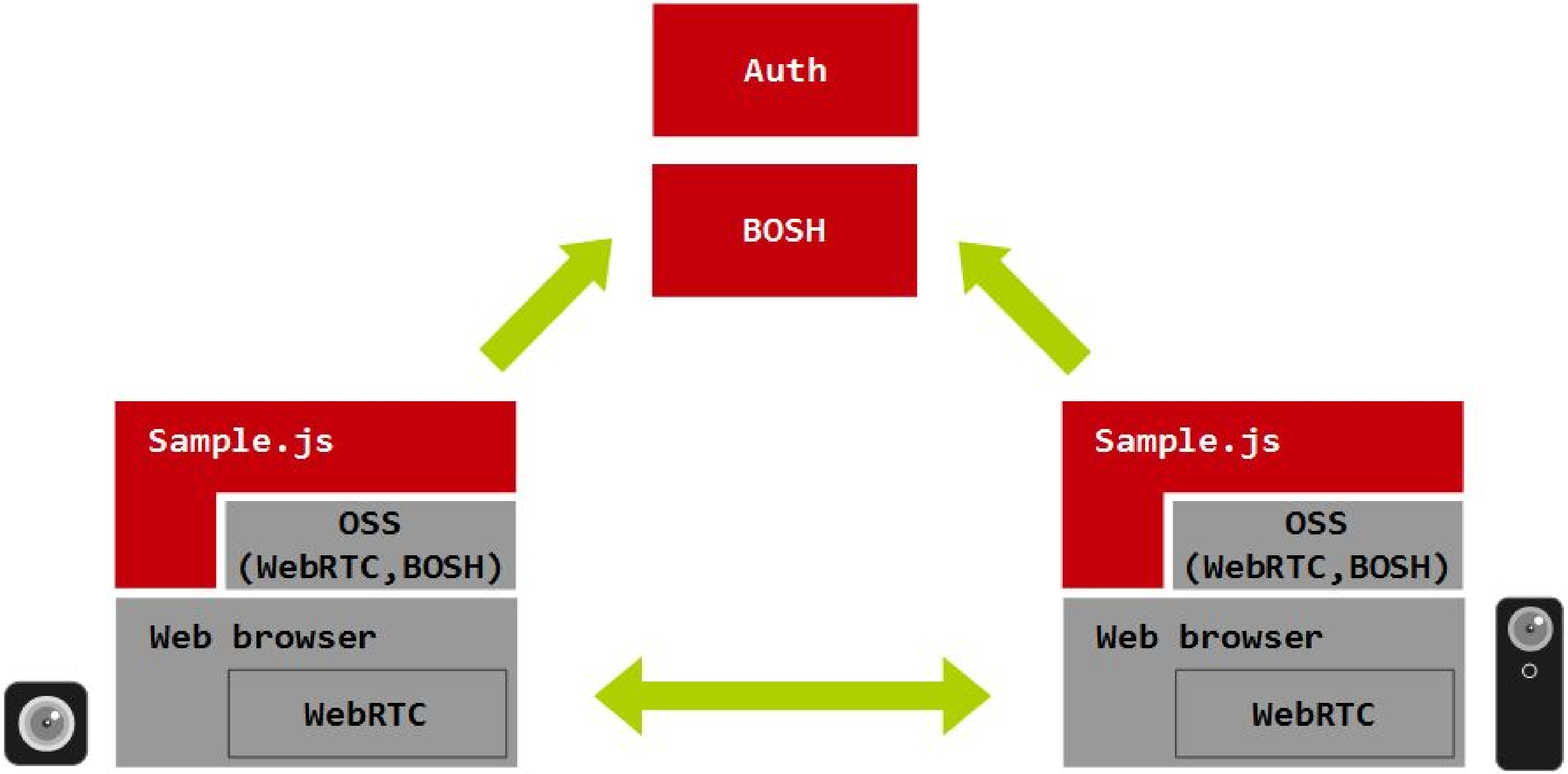
Streaming Functionality

- Overview
  - Video chat (no audio) between 2 PCs
  - Uses WebRTC (Web browser functionality)



Streaming Functionality

- Architecture
  - Auth/BOSH (Bidirectional-streams Over Synchronous HTTP) Server
  - JavaScript/HTML



Usage

- Clone the sample code

```
$ git clone https://github.com/ricohapi/video-streaming-sample-app
```

- Configure client credentials

```
$ cd video-streaming-sample-app
$ cp samples/config_template.js samples/config.js
$ vi samples/config.js
```

- Build

```
$ npm install
$ gulp build
```

- Run

```
$ gulp run
```

https://github.com/ricohapi/video-streaming-sample-app

Summary

- Uses signaling server
  - WebRTC(over BOSH)
- OK to use one user ID
  - abc@example.com+mac
  - abc@example.com+theta
- View sample similar to theta360.com
  - WebGL

Please think of interesting usage examples
Ideas and feedback welcome

3. Photo and Media Storage

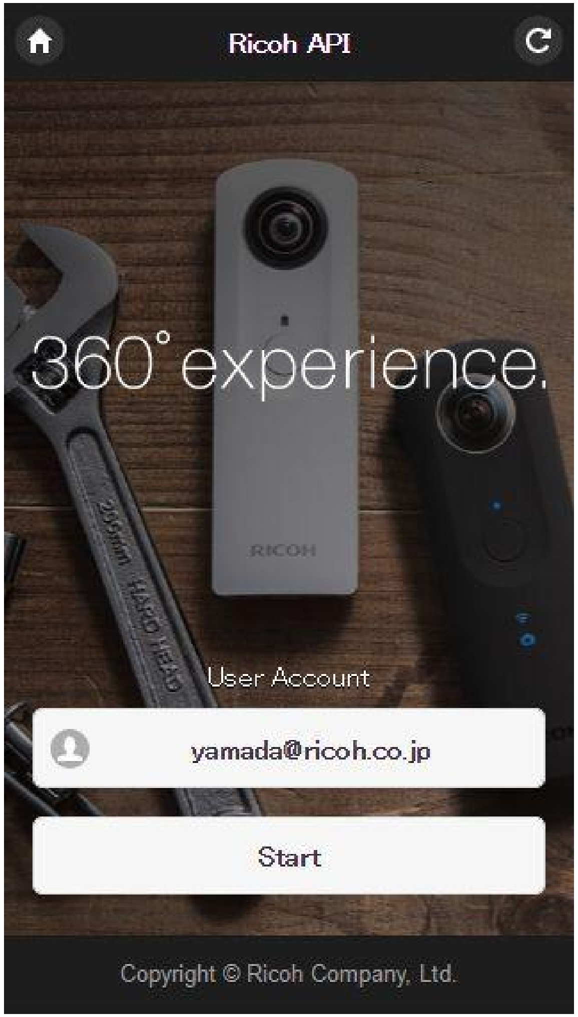
Originally presented by
Hideshi Hosono
RICOH New Technology Development HQ
SV Technology Development Center, Platform Development Lab

Basic Storage API

Storage Function	**REST API**	**SDK API**
Uploading Media	POST /media	.upload()
Get List of Saved Media	GET /media	.list()
Get Media Information	GET /media/{id}	.info()
Get Media Data	GET /media/{id}/content	.download()
Delete Media	DELETE /media/{id}	.delete()
Service URL: https://mss.ricohapi.com/v1/media		

Authentication / Authorization	**REST API**	**SDK API**
User Authentication	POST /auth/token	.connect()
Service URL: https://auth.beta2.ucs.ricoh.com/auth/token		

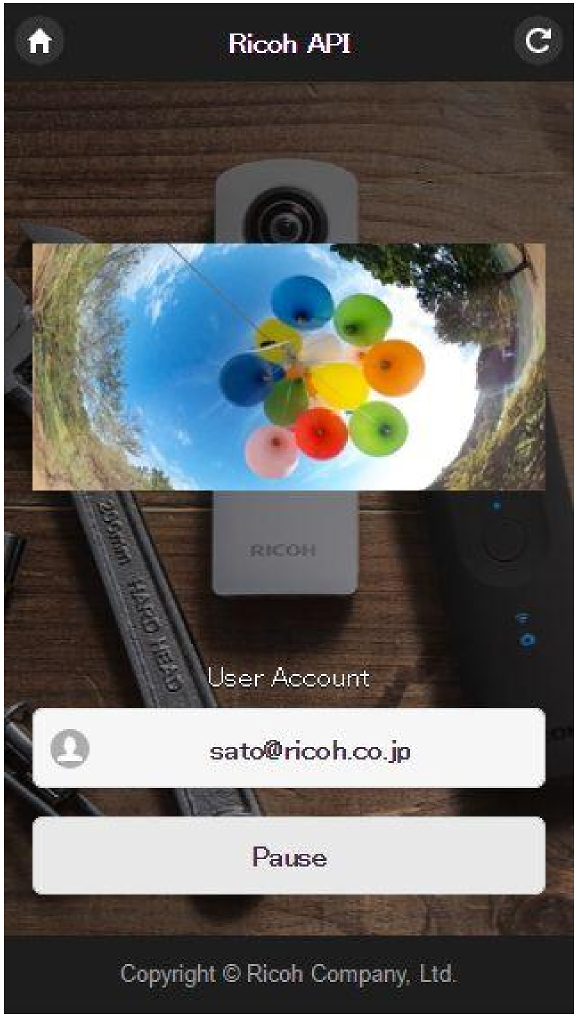
Try and Create Quick Apps



User Account

yamada@ricoh.co.jp

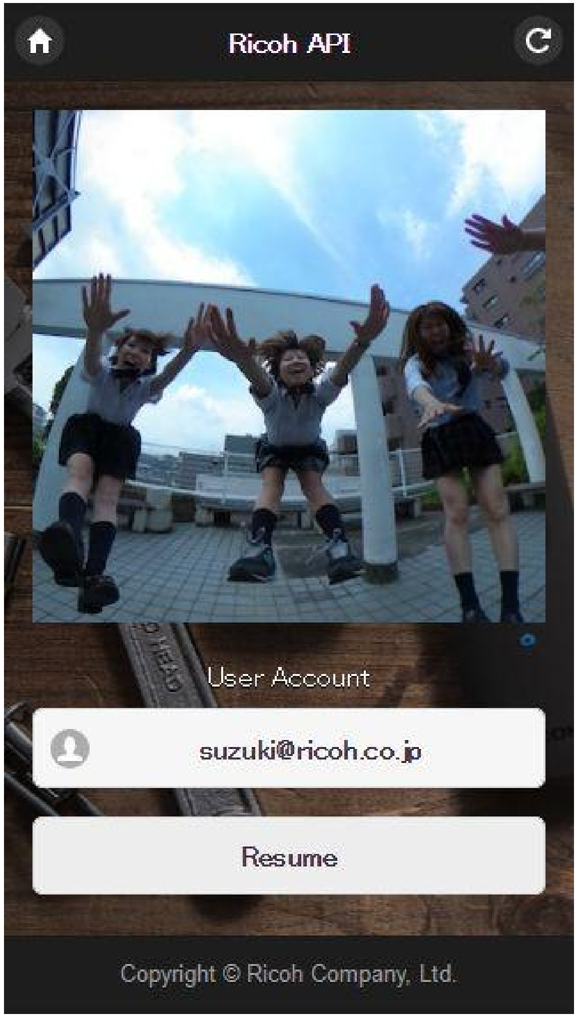
Copyright © Ricoh Company, Ltd.



User Account

sato@ricoh.co.jp

Copyright © Ricoh Company, Ltd.



User Account

suzuki@ricoh.co.jp

Copyright © Ricoh Company, Ltd.

Quickly created storage applications

Summary of Demo App

Benefit 1: Cloud Compatibility

Using the RICOH Cloud Media Storage Service, you can do slideshows of your stored media

Benefit 2: Multiple Accounts

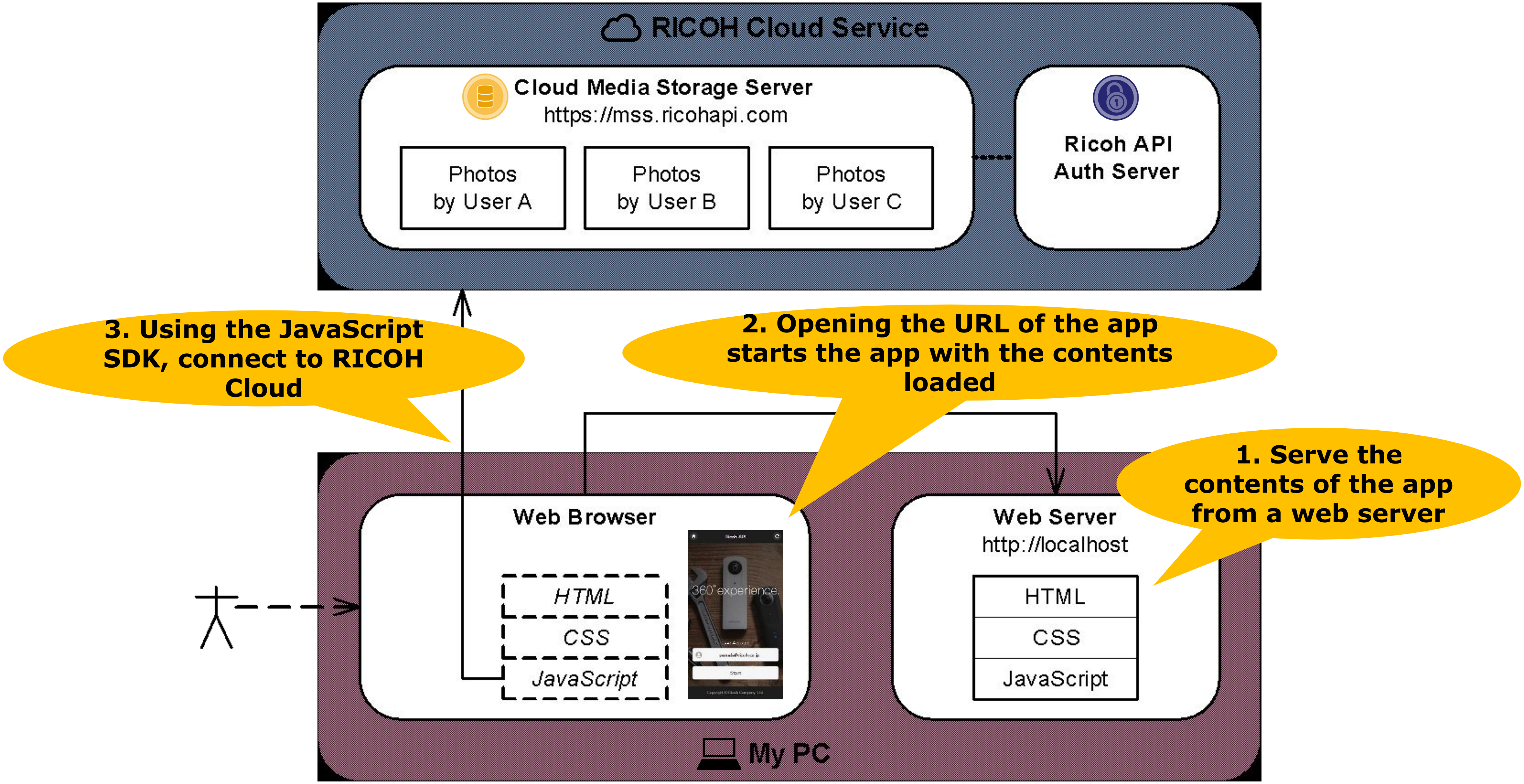
Switching between user accounts is possible

Benefit 3: Multi-platform

Build in HTML5 and run on Mac OS, Windows, iOS, Android

Let’s try running the app in a browser

Full Structure of System



RICOH Cloud Service

Cloud Media Storage Server
https://mss.ricohapi.com

Photos by User A

Photos by User B

Photos by User C

Ricoh API Auth Server

3. Using the JavaScript SDK, connect to RICOH Cloud

2. Opening the URL of the app starts the app with the contents loaded

1. Serve the contents of the app from a web server

Web Browser

HTML

CSS

JavaScript

Web Server
http://localhost

HTML

CSS

JavaScript

My PC

If you try to build a demo app...

Doing security authentication in the cloud is a pain in the neck but…

Using the SDK we produced an app with around 100 lines of code

Source Code Structure

File	Contents	Lines of Code
index.html	HTML file	41 lines
ric-sample.css	Stylesheet	25 lines
ric-sample.js	JavaScript file	53 lines
ricohapi-mstorage.js	Media Storage SDK	(On GitHub)
Total		**~100 lines !** (119 lines)
※line count does not include blank lines、comments、logs		

Let’s try testing the source code in the browser

Program Processing Flow

- ✓ Start/Pause/Resume button tap、ric-sample.js event handler function is called

  Start:
  - ✓ List images saved in cloud
  - ✓ Timer started

  Pause:
  - ✓ Timer stopped

  Resume:
  - ✓ Timer started

- ✓ In timer processing, image is downloaded from the Cloud and displayed at regular intervals

Cloud Transfer

2 Points

Transfer Data With The Cloud

Cloud Transfer Point 1

Point 1

Before you connect to the cloud、set the authentication information through the SDK API

Then use the SDK ! When you transfer data with the cloud server, the SDK will handle the security authentication.

※Authentication Information: Client ID/Secret, User ID/Password

Continuation of Point 1

```
$(document).ready(function() {
    var authClient = new AuthClient("koTjKwBm…", "rWOwyT5d…"),
        mediaStorage = null;
    …
    $("#ric-start-stop").on("click", function() {
        var buttonText = $(this).text();
        if (buttonText == "Start") {
            var userID = $("#ric-account").val();
            authClient.setResourceOwnerCreds(userID, USER_PASSWORD[userID]);

            mediaStorage = new MStorage(authClient);
            mediaStorage.connect();
            …
        }
        …
    });
});
```

1. **When starting the app set the Client ID/Secret**

2. Start Button pressed, User ID/Password auth

3. When starting to use the Storage API set the authentication information

When using the Storage API the SDK will automatically do security authentication

Give it and try and test it in your browser

Cloud Transfer Point 2

Point 2

Since image data that is stored in the cloud will be a security problem, you cannot directly reference the <img> tag.

We recommend the technique of getting data from the cloud securely and converting it into Binary Large OBject (Blob) format.

Image data in Blob format allows referencing the <img> tag by creating an object URL in memory.

Leave data acquisition and conversion up to the SDK!
The SDK supports image data in Blob format.

Point 2, continuation

```
function slideshow() {
    var mediaID = photoList[photoIndex++].id;
    if (photoIndex == photoList.length) photoIndex = 0;

    mediaStorage.download(mediaID, "blob")
        .then(function(blob) {
            var imageURL = URL.createObjectURL(blob);
            $("#ric-view")
                .one("load", function() { URL.revokeObjectURL(imageURL); })
                .attr("src", imageURL);
        });
    …
}
```

1. Assign Blob format get media data

2. Stored Blob data creates object URL

3. Set <img> tag src to object URL

Display cloud image

Give it and try and test it in your browser

Additional Information

Porting to the browser SDK

The Media Storage JavaScript SDK uses Node.js

In order to use it in a browser, you'll need to modify the source code and use something like Browserify.

We have plans to distribute the modified source code in our GitHub repository.

Media Storage SDK for Browser:

https://github.com/ricohapi/media-storage-js/tree/master/build

※plan to distribute soon

4. Q&A

5. Future Ideas

Future Ideas

The following functionality is under review

・Remote Control

・Sensor Support

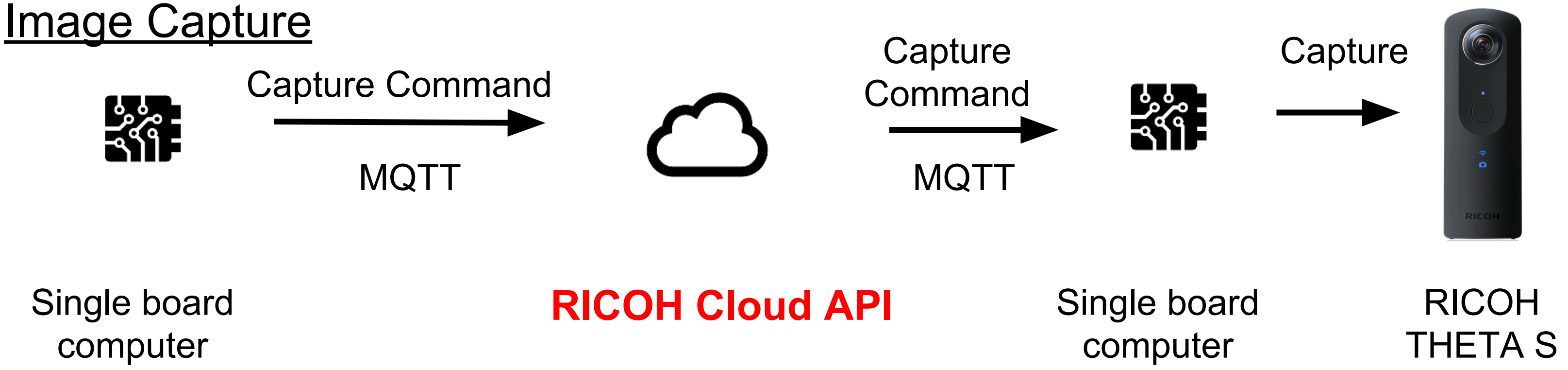
・Image processing, image recognition

・Data aggregation、Analysis

・SNS Support

Idea For Remote Control

Example 1, Image Capture



Single board computer

Capture Command

MQTT

RICOH Cloud API

Capture Command

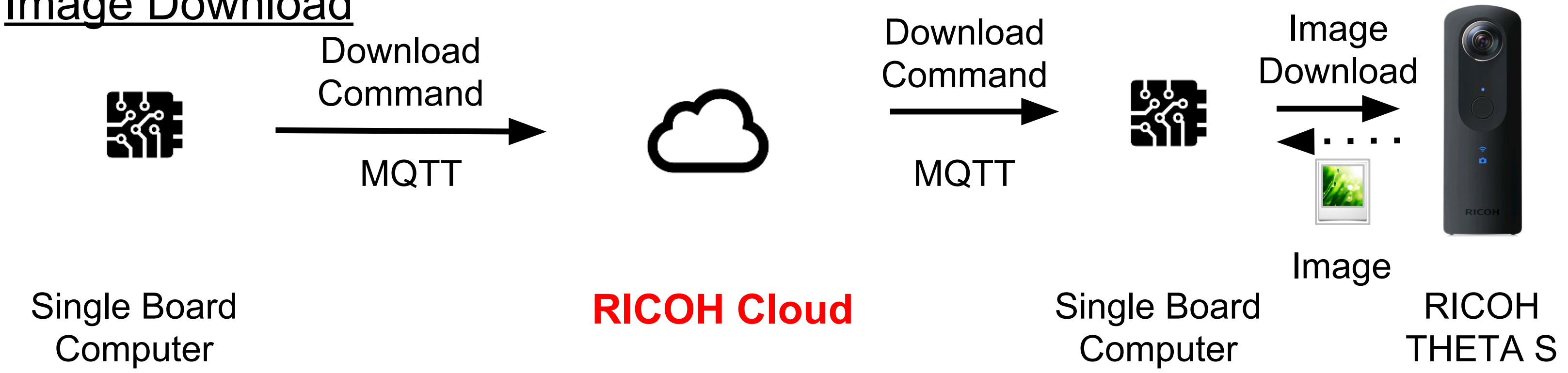
MQTT

Single board computer

Capture

RICOH THETA S

Example 2, Image Download



Single Board Computer

Download Command

MQTT

RICOH Cloud

Download Command

MQTT

Single Board Computer

Image Download

Image

RICOH THETA S

Summary

About the Cloud API

- Features
- Usage

Usage

- Video Communication
- Photo and Media Storage

Future Plans and Ideas

Enjoy RICOH THETA and RICOH Cloud API!!