SDK6 AN Remote Command Control Client

Version 0.1

February 12, 2015

Confidentiality Notice:

US
3101 Jay Street
Ste.110
Santa Clara, CA 95054, USA
Phone: +1.408.734.8888
Fax: +1.408.734.0788

Taiwan
Suite C1, No. 1, Li-Hsin Road 1
Science-Based Industrial Park
Hsinchu 30078, Taiwan
Phone: +886.3.666.8828
Fax: +886.3.666.1282

Hong Kong
Unit A&B, 18/F, Spectrum Tower
53 Hung To Road
Kwun Tong, Kowloon
Phone: +85.2.2806.8711
Fax: +85.2.2806.8722

Japan - Yokohama
Shin-Yokohama Business Center Bldg. 5th Floor
3-2-6 Shin-Yokohama, Kohoku-ku,
Yokohama, Kanagawa, 222-0033, Japan
Phone: +81.45.548.6150
Fax: +81.45.548.6151

Korea
6 Floor, Hanwon-Bldg.
Sunae-Dong, 6-1, Bundang-Gu
SeongNam-City, Kyunggi-Do
Republic of Korea 463-825
Phone: +031.717.2780
Fax: +031.717.2782

China - Shenzhen
Unit E, 5th Floor
No. 2 Finance Base
8 Ke Fa Road
Shenzhen, 518057, China
Phone: +86.755.3301.0366
Fax: +86.755.3301.0966

China - Shanghai
9th Floor, Park Center
1088 Fangdian Road, Pudong New District
Shanghai 201204, China
Phone: +86.21.6088.0608
Fax: +86.21.6088.0366

I Contents

II Preface

This document provides technical details using a set of consistent typographical conventions to help the user differentiate key concepts at a glance.

Conventions include:

Example	Description
AmbaGuiGen, **DirectUSB** **Save, File > Save** **Power, Reset, Home**	Software names GUI commands and command sequences Computer / Hardware buttons
Flash_IO_control **da**, **status**, **enable**	Register names and register fields. For example, **Flash_IO_control** is the register for global control of Flash I/O, and bit 17 (**da)** is used for DMA acknowledgement.
GPIO81, **CLK_AU**	Hardware external pins
VIL, VIH, VOL, VOH	Hardware pin parameters
INT_O, RXDATA_I	Hardware pin signals
amb_performance_t ***amb_operating_mode_t*** **amb_set_operating_mode()**	API details (e.g., functions, structures, and type definitions)
`/usr/local/bin` `success = amb_set_operating_mode (amb_base_address, & operating_mode)`	User entries into software dialogues and GUI windows File names and paths Command line scripting and Code

Table II-1. Typographical Conventions for Technical Documents.

Additional Ambarella typographical conventions include:

- Acronyms are given in UPPER CASE using the default font (e.g., AHB, ARM11 and DDRIO).
- Names of Ambarella documents and publicly available standards, specifications, and databooks appear in *italic* type.

1 Overview

1.1 Overview: Introduction

The Remote Command Control (hereinafter referred to as RCC) is to control the Ambarella camera via network, such as WiFi or BlueTooth. The command set is defined in JavaScript Object Notation (JSON) format. For more details, please refer to *AMBARELLA ASeries Wireless Connectivity API Remote Control* document.

There are two roles in RCC operations, namely client and camera. The client could be the APP on phone, PC or any other devices which could perform the network connection. In the normal operations, the client will send the request command to the camera and the camera will give the response of the operation result to the client. If the operation of the remote command is blocked, the client cannot issue another command before the camera sends back a response (during command execution), again. If the client sends out a request before the camera responses to the previous request, the request will be ignored. There will be more operation flows in chapter 2.

Before starting the RCC flow, the client has to connect with the camera through a dedicated network interface. The connecting protocol might need to be changed for different interfaces. For example, if the client connects to camera with WiFi, it will use the TCP socket. However, if the client connects camera with Bluebooth, it will use the RFCOMM port. The default protocol for a dedicated network interface is explained below:

Network type	Command Service	Data Service
WiFi	TCP port 7878	TCP port 8787
Bluetooth (BT)	RFCOMM port 20	RFCOMM port 10
Bluetooth Low Energy (BLE)	Command: 11111111-616d-6261-5f69-645f62617365 Response: 33333333-616d-6261-5f69-645f62617365	None

Table 1-1. Relationship between Network Type, Command Service and Data Service.

RCC comprises of commands and data service. The command service is responsible for receiving requests, sending responses and notifications with JSON on different interfaces; and data service is responsible for data transmission on different interfaces. These services will act as a daemon in order to accept client requests. The client must establish a wireless connection before it sends out any request commands. All requests are issued by the client in the active role. All responses or notifications are issued by the service in the passive role. The entire flow of each operation is where the client sends a out request command and receives a response as the camera sends out the information. If notification is necessary, the camera also sends out notification as well.

2 The Sequential Diagram of Operation

The application in all pictures represents the client which might have connection ability. Every request from the client must have a response which is sent out by the camera. The camera can only be controlled with an unique client, but it does not allow two clients to control it at the same time.

2.1 The Sequential Diagram of Operation: Normal Operations

The application establishes a connection with the camera on a specific port for request, response and notification before the session starts. Every application should get the session from the camera first before any operations are performed. After the session starts, the application will establish a connection with the camera on a specific port for data transmission.

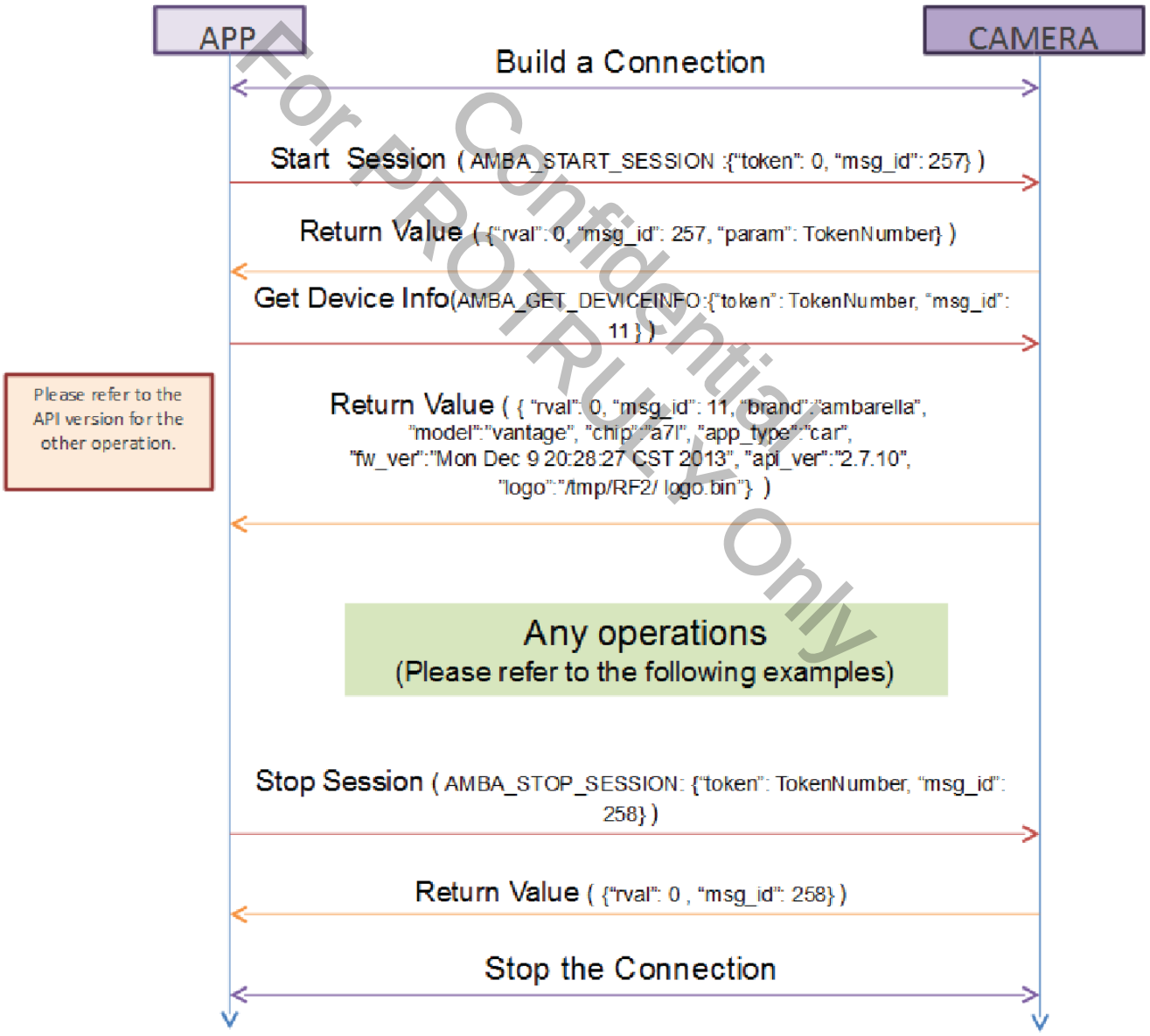

Figure 2-1. The Sequential Diagram of Normal Operations.

If the application has gotten the session from the Camera, the following are some of the examples which can be operated.

2.1.1 Normal Operations: Take a Photo

Using the Command: **AMBA_TAKE_PHOTO**
Arguments: {“token”: TokenNumber, “msg_id”: 769}
Successful return: {“rval”: 0, “msg_id”: 769, “param”:
“/tmp/fuse_d/DCIM/100MEDIA/AMBA0005.jpg”}

2.1.2 Normal Operations: Record a Clip

Using the Command: **AMBA_RECORD_START** and **AMBA_RECORD_STOP**
Start the record:
Arguments: {“token”: TokenNumber, “msg_id”: 513}
Successful return: {“rval”: 0, “msg_id”: 513}
Stop the record:
Arguments: {“token”: TokenNumber, “msg_id”: 514}
Successful return: {“rval”: 0, “msg_id”: 514,
“param”:“/tmp/fuse_d/DCIM/100MEDIA/AMBA0001.mp4”}

2.1.3 Normal Operations: Force Split

Using the Command: **AMBA_FORCE_SPLIT**
(The recorder has to start before the application sends this command)
Arguments: {“token”: TokenNumber, “msg_id”: 516}
Successful return: {“rval”: 0, “msg_id”: 516, “param”:
“/tmp/fuse_d/DCIM/100MEDIA/AMBA0005.mp4”}

2.2 The Sequential Diagram of Operation: Set Settings

If the application wants to change the setting, it should stop the view finder or recording first.

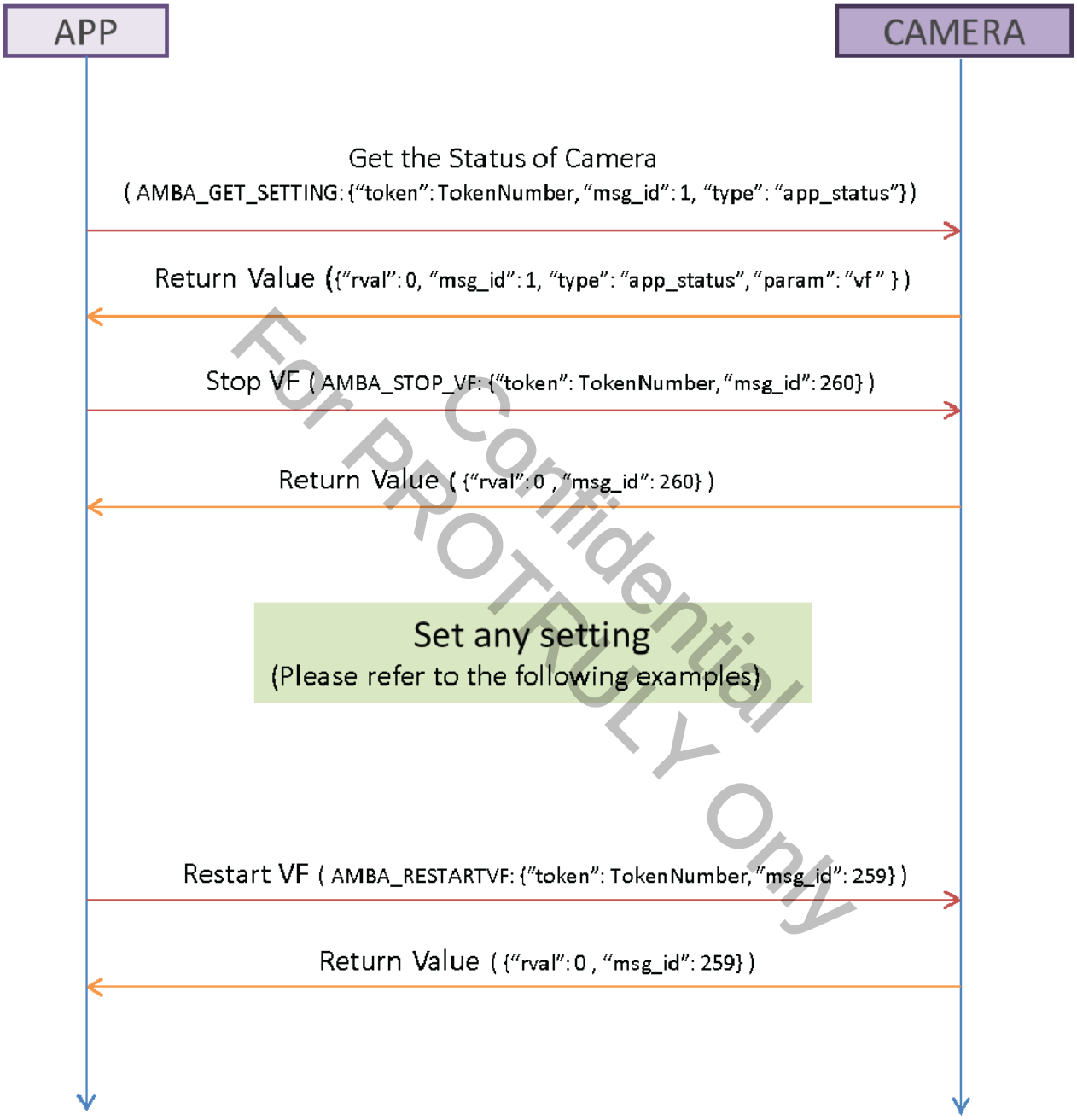

Figure 2-2. The Sequential Diagram of Setting when the Current Status is View Finder.

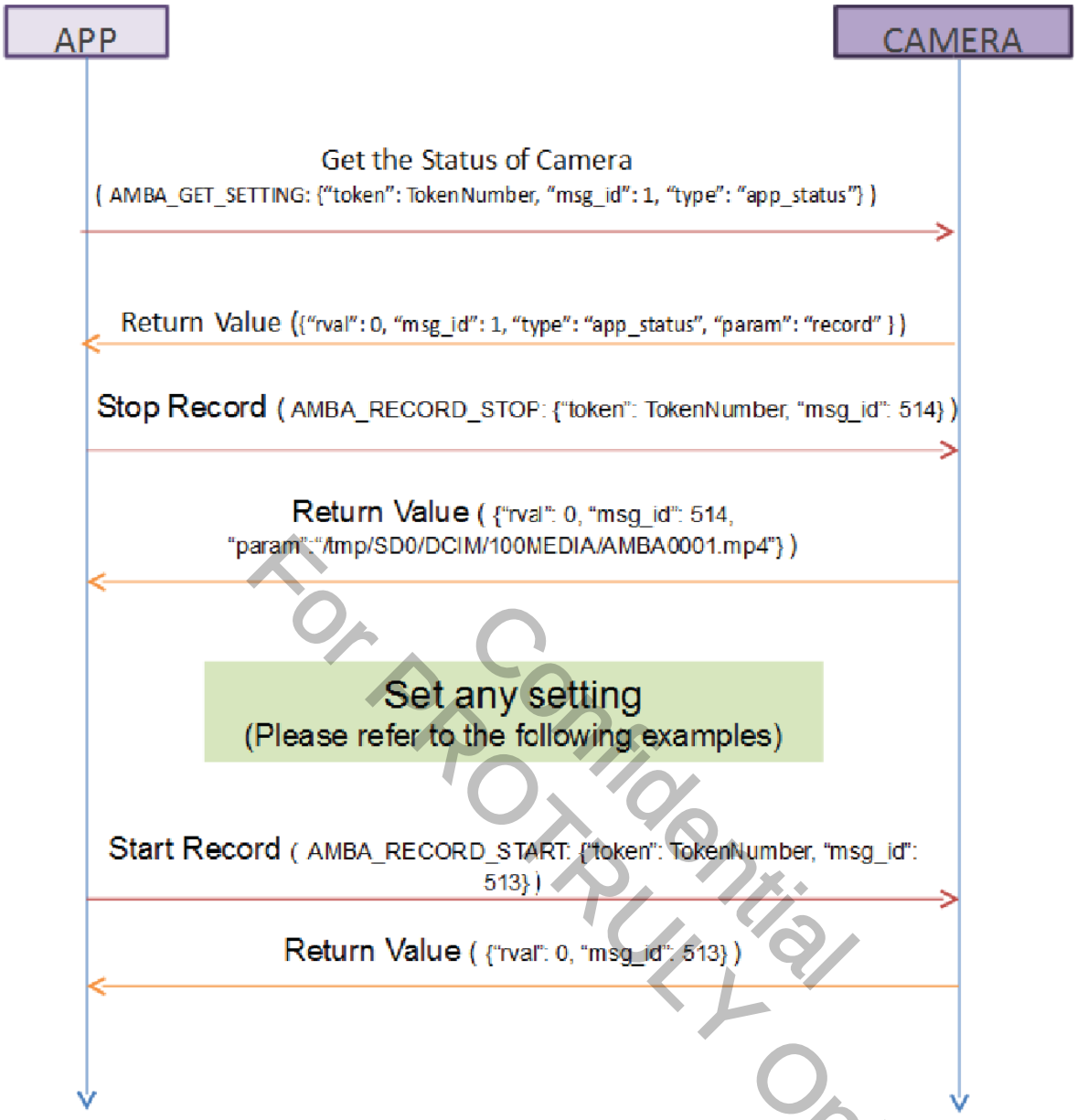

Figure 2-3. The Sequential Diagram of Setting when the Current Status is Recording.

If the VF or recorder has been stopped, the application can perform any setting as stated in the examples below:

2.2.1 Set Settings: Set the Setting of Streaming

If the API version < 2.8.0, the application should disable **std_def_video**, set the stream_type to RTSP and Enable stream_while_record while using the command **AMBA_SET_SETTING**.
Arguments to Disable Standard Definition Video:
{“token”: TokenNumber, “msg_id”: 2, “type”: “std_def_video”, “param”: “off”}

Arguments for setting Viewfinder to Type RTSP:
{“token”: TokenNumber, “msg_id”: 2, “type”:“stream type”, “param”: “rtsp”}

Arguments to Enable view finder Transmit While Recording:
{“token”: TokenNumber, “msg_id”: 2,“type”: “stream_while_record”, “param”: “on”}
If the API version >= 2.8.0, the App should set two settings (save_low_resolution_clip and stream_out_type) with using the command “AMBA_SET_SETTING”.

Arguments to Disable Standard Definition Video:
{“token”: TokenNumber, “msg_id ”: 2,“type”: “save_low_resolution_clip”, “param”: “off”}

Arguments for setting Viewfinder to Type RTSP:
{“token”: TokenNumber, “msg_id ”: 2,“type”: “stream_out_type”, “param”: “rtsp”}
On successful command operations of this example, an RTSP stream can be retrieved over WiFi from: rtsp://<Camera IP>/live.

(Please refer to the document *AMBARELLA Application Note: Streaming Settings for Remote Control APIs for Versions later than v2p8p0 (General)* for more details)

2.2.2 Set Settings: Set the Resolution

Using the command **AMBA_SET_SETTING** with type is “video_resolution” and specified resolution.
Arguments for video resolution 1280x720p60: {“token”: TokenNumber, “msg_id”: 2, “type”: “video_ resolution”, “param”: “1280x720 60P 16:9”}
Successful return: {“rval”: 0 , “msg_id”: 2, “type”: “video_resolution”}

2.3 The Sequential Diagram of Operation: Query Session Holder

The camera will send out this query command to the client who currently holds the session (i.e., has locked the token) in the event that another client tries to create a new session. When this command is received, a timeout period begins. During this period, the client who holds the session is obligated to respond with its token in order to indicate that the session is still alive. Otherwise, the camera will close the existing session and allow the creation of a new session once the timeout period has elapsed. The default timeout period is 800 ms.

The following sequential diagrams explain the above sequences.

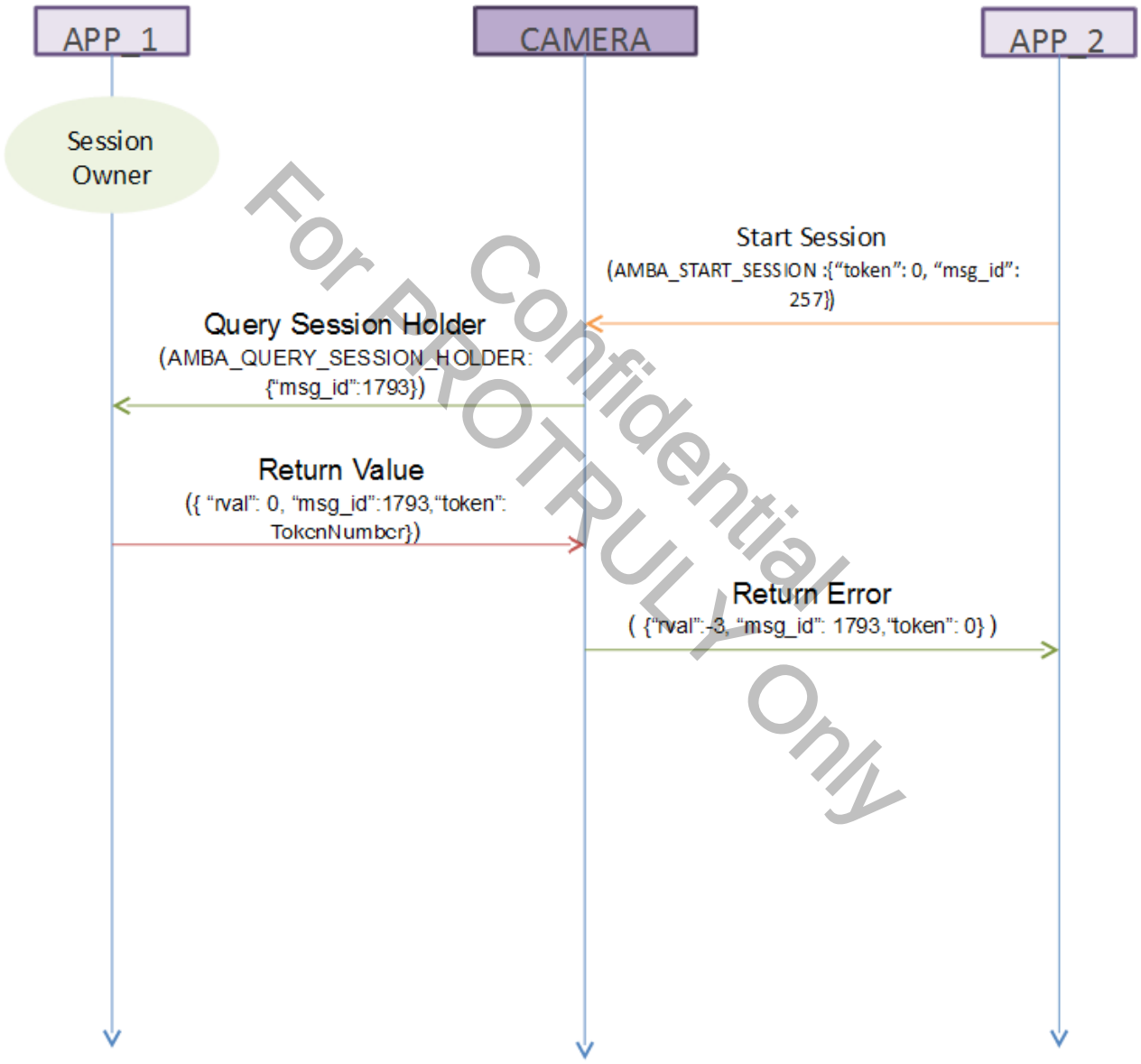

*Figure 2-4. The Sequential Diagram of the Command **AMBA_QUERY_SESSION** is Successful.*

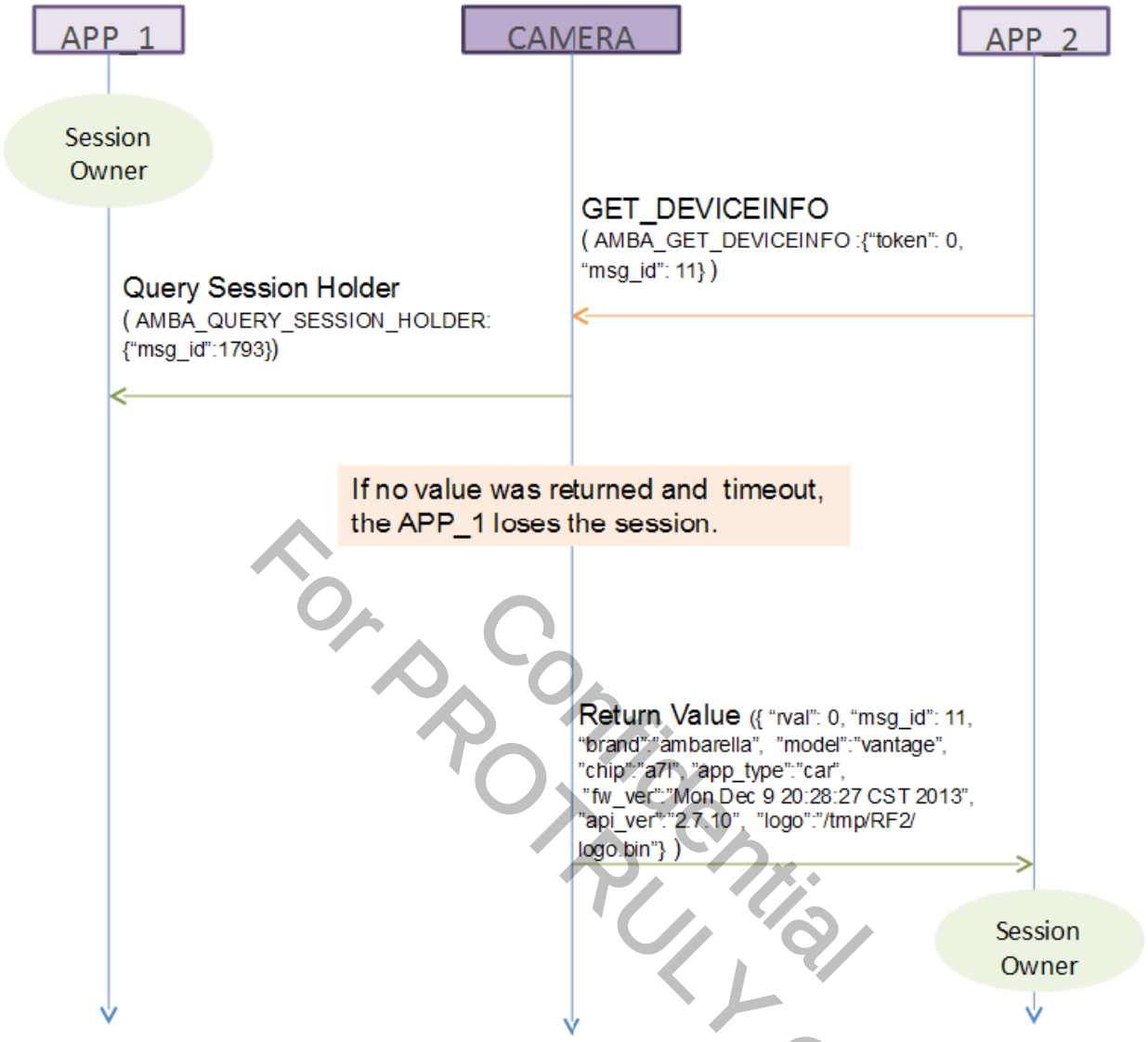

*Figure 2-5. The Sequential Figure of the Command **AMBA_QUERY_SESSION** Fails.*

2.4 The Sequential Diagram of Operation: Get Single Setting

The generalized API (**AMBA_GET_SETTING**) is used to retrieve the current value for a single camera parameter within the camera parameter database. This API can be used for all camera parameters.
The following example is the sequential diagram for getting the current video resolution.
(Please refer to the document *AMBARELLA A-Series Wireless Connectivity API: Remote Control* for more details)

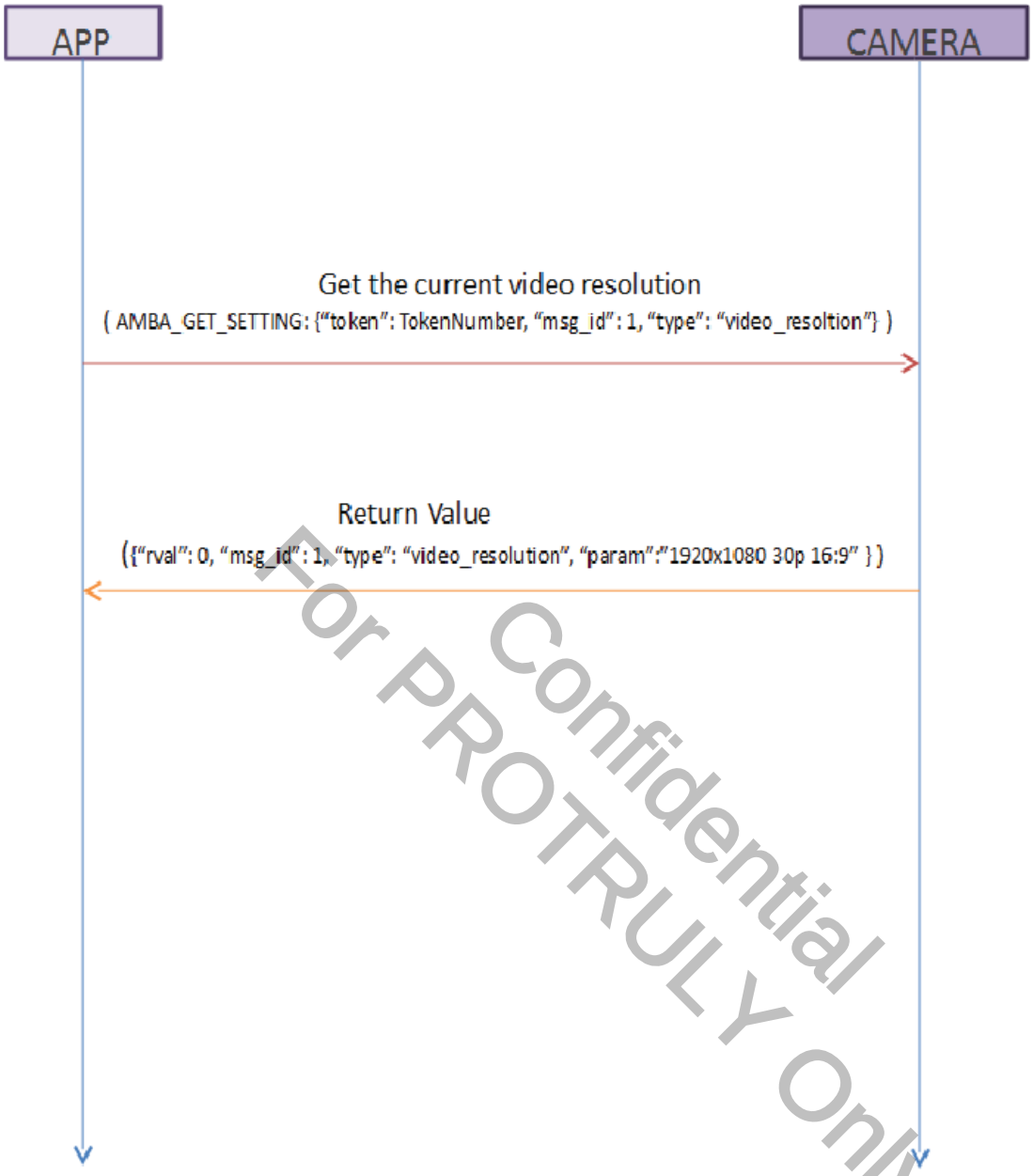

Figure 2-6. The Sequential Diagram of Getting the Current Video Resolution.

2.5 The Sequential Diagram of Operation: Get All Current Settings

The generalized API (**AMBA_GET_CURRENT_SETTINGS**) is used to retrieve current values for all camera parameters. This API can be used for all camera parameters.

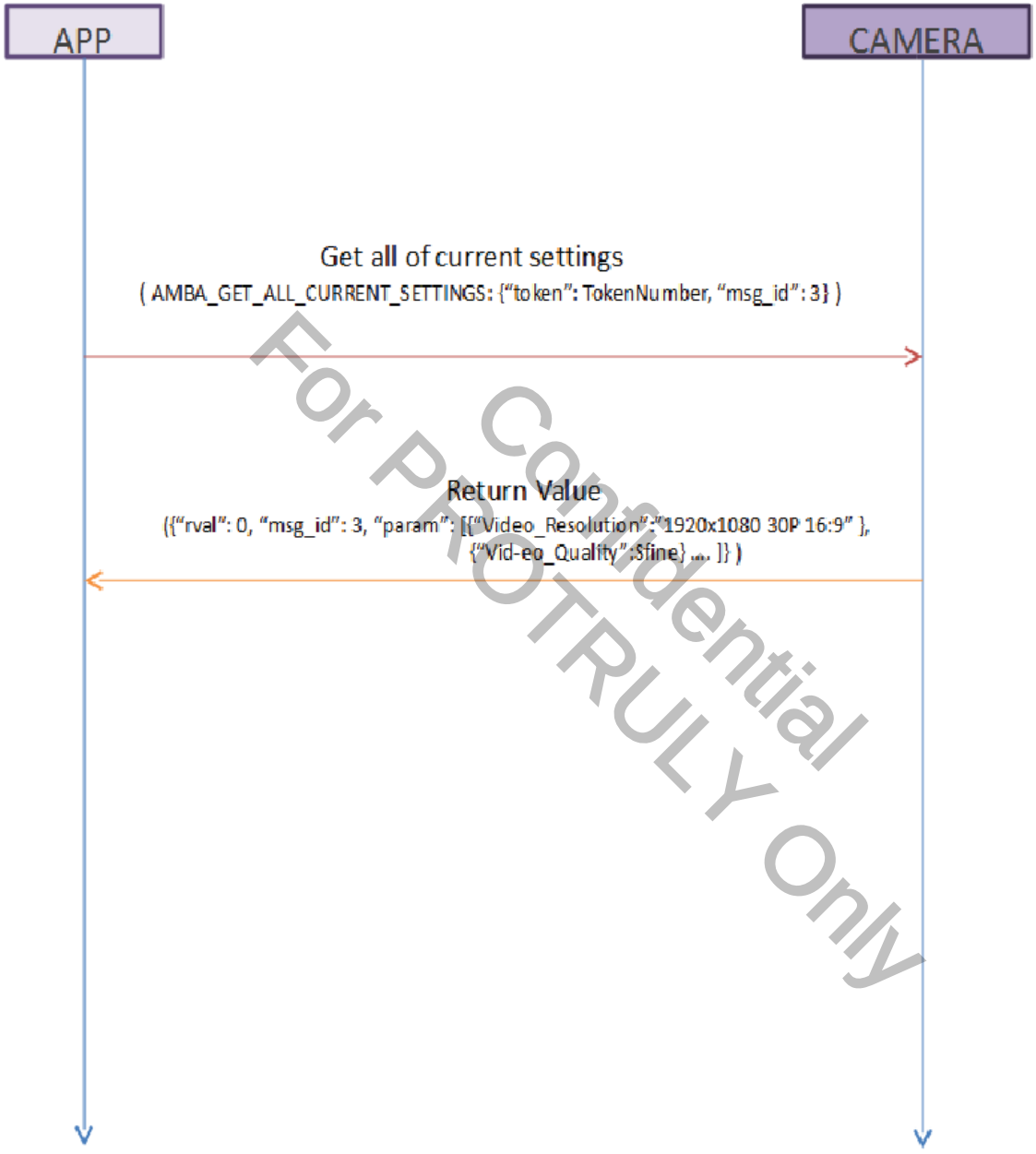

Figure 2-7. The Sequential Diagram of Getting All of the Current Settings.

2.6 The Sequential Diagram of Operation: Set Active Client Information

This API (**AMBA_SET_CLNT_INFO**) sets the address information of the active client. The handheld device is obligated to set address information before creating the data socket connection. The camera will verify the address information with the incoming data socket connection and reject it if verification fails.

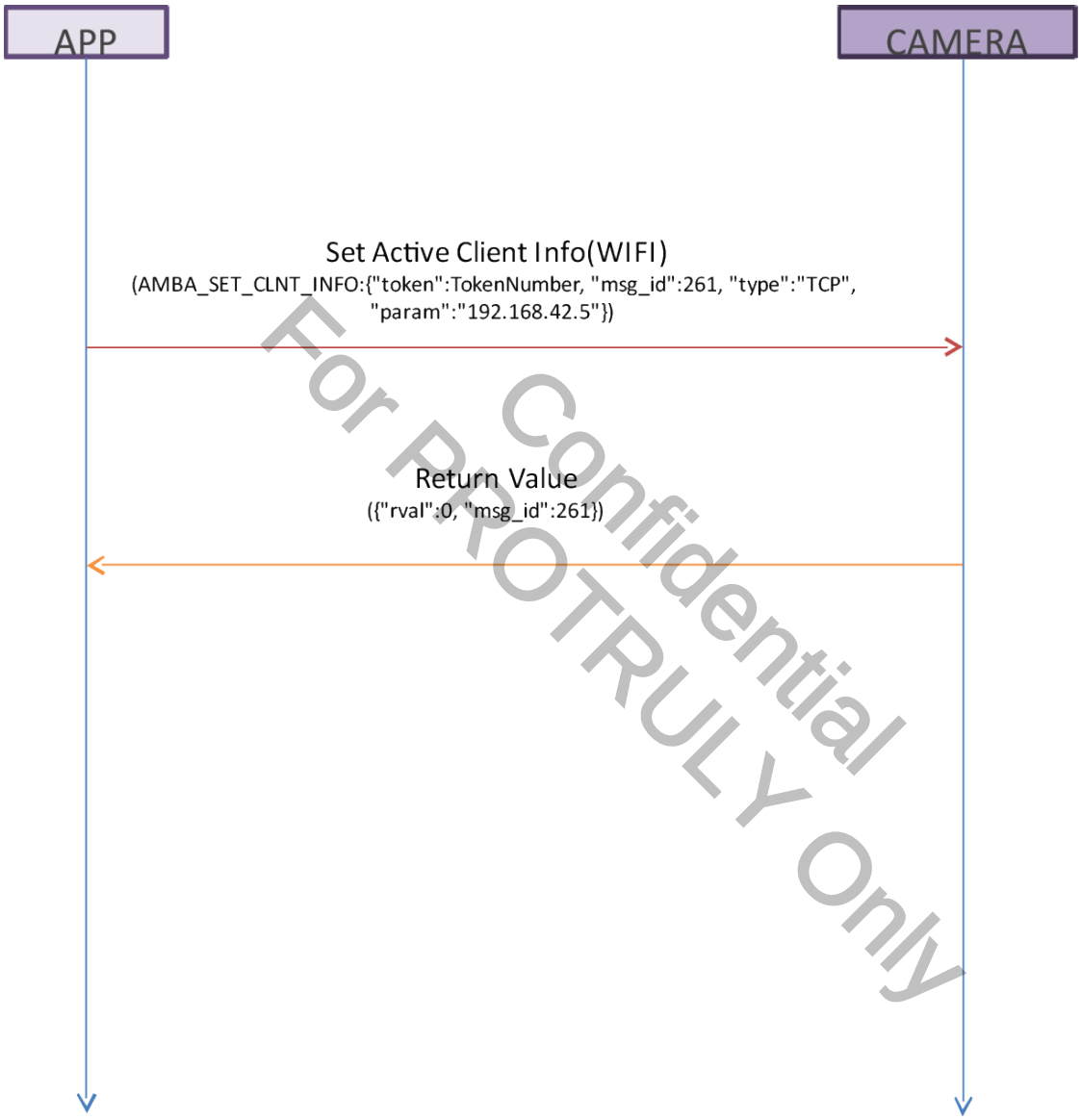

Figure 2-8. The Sequential Diagram of Setting Active Client Information by Using WIFI.

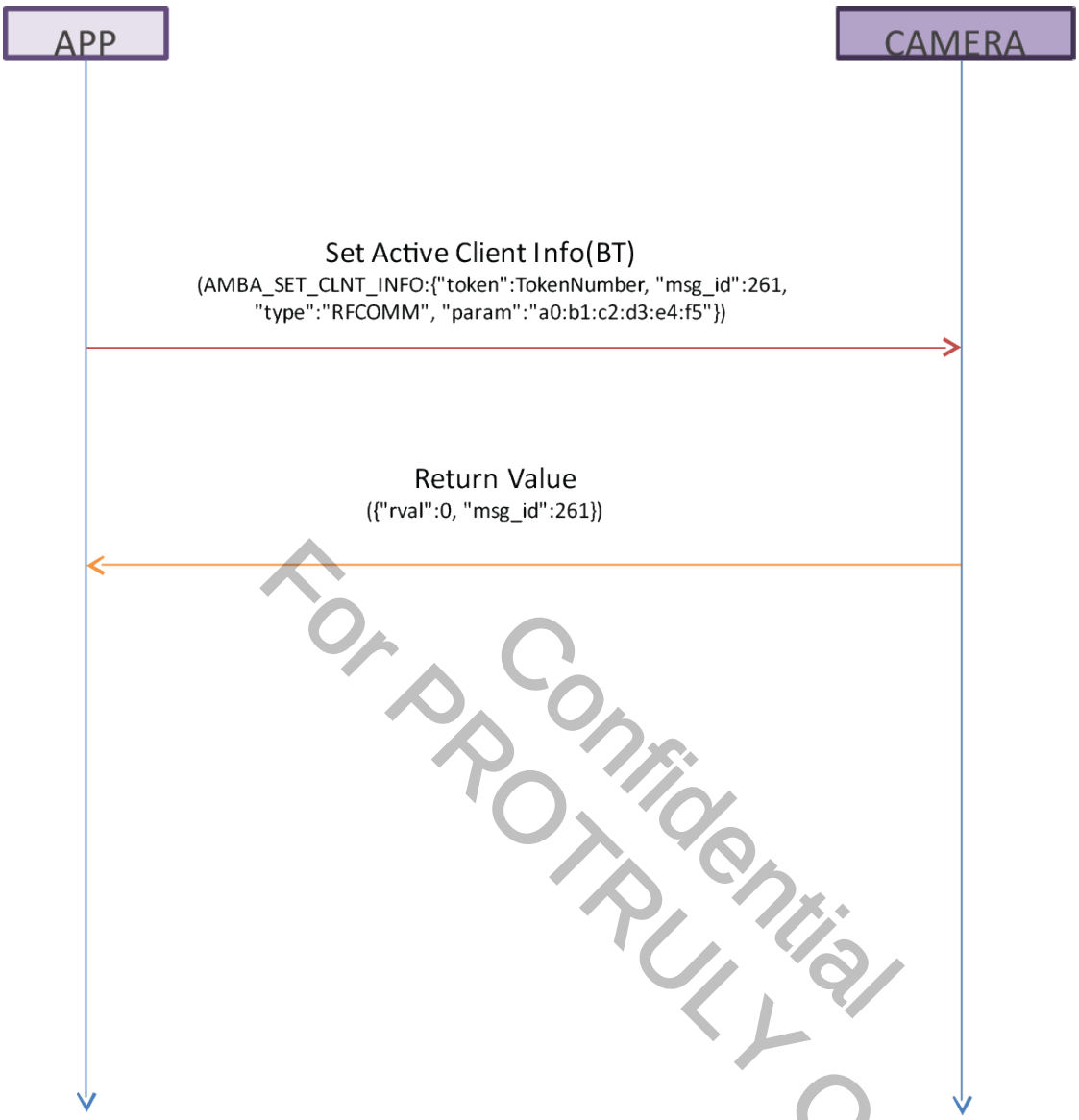

Figure 2-9. The Sequential Diagram of Setting Active Client Information by Using BT.

2.7 The Sequential Diagram of Operation: Get File

The remote device could use **AMBA_GET_FILE** to download the multimedia file and then play it or upload it to the social network. The transferring port for files is 8787. If camera receives the command **AMBA_GET_FILE** and the file is found, camera transmits data to the client who has connected to the port 8787. Currently, only one file which can be transferred at a time is allowed.

Note: Send **AMBA_SET_CLNT_INFO** command to specify client information before **AMBA_GET_FILE** starts.

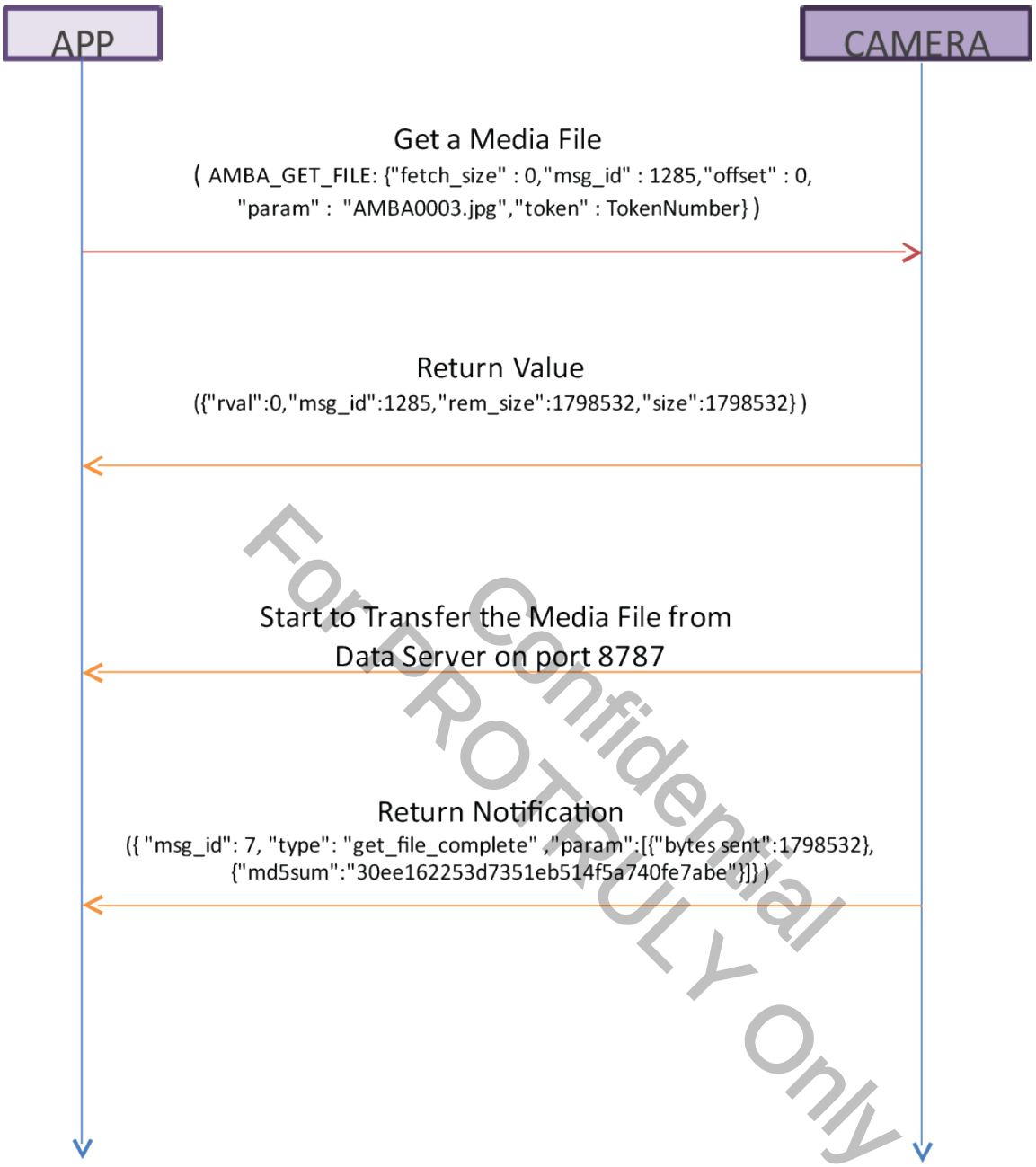

Figure 2-10. The Sequential Diagram of Getting a Media File.

2.8 The Sequential Diagram of Operation: Get the Thumbnail

The API (**AMBA_GET_THUMB**) can be used to retrieve a thumbnail file. The user should not send any command until the size of the returned values which matches the transmitted size of data. The thumbnail and IDR (Instantaneous Decoding Refresh, that is used on the video) are relatively small in size and should take very little time to transfer. Therefore, the camera does not send any notification after the period of transmitting the thumbnail or IDR was completed.

Note: Send **AMBA_SET_CLNT_INFO** command to specify client information before **AMBA_GET_THUMB** starts.

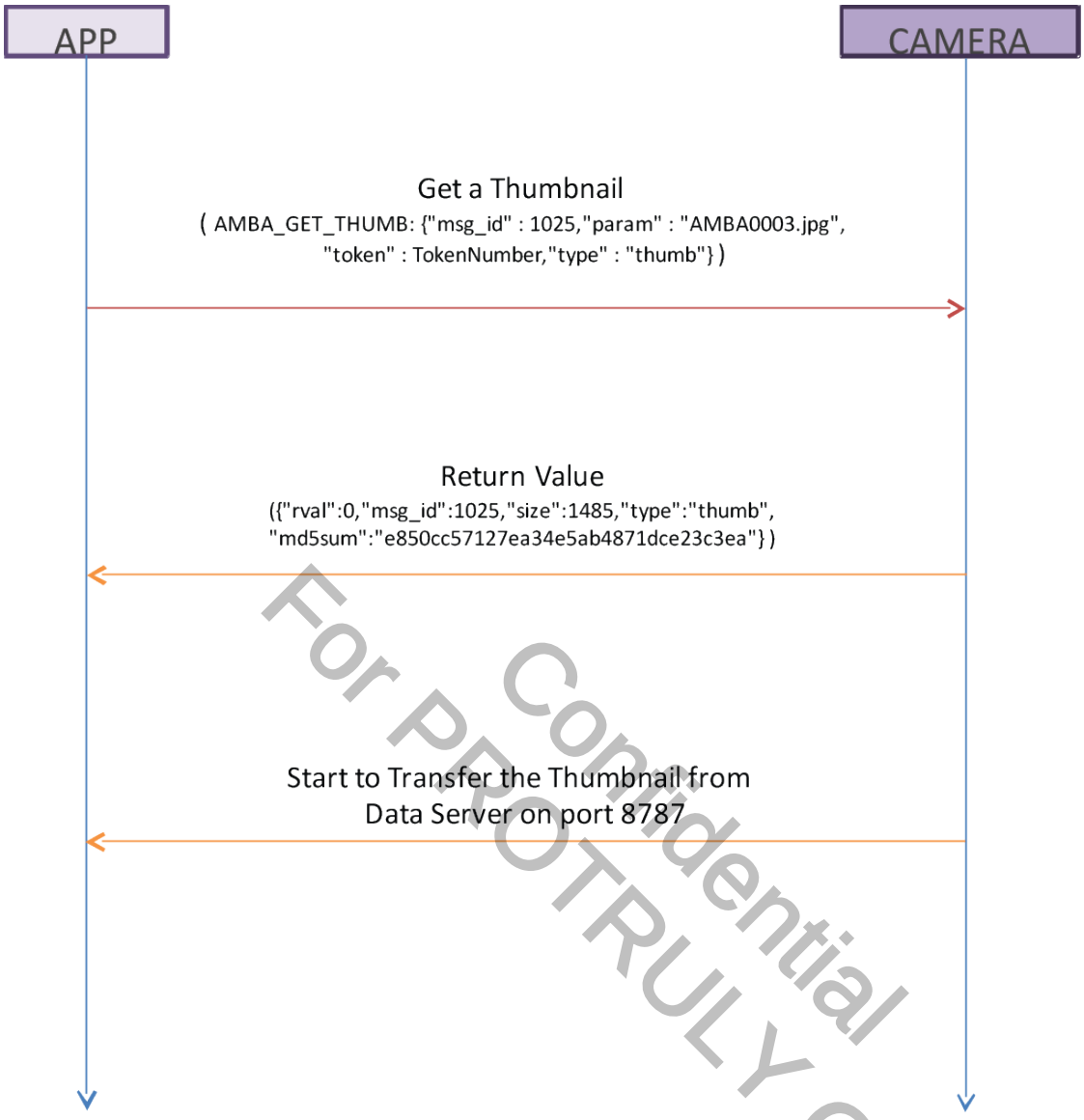

Figure 2-11. The Sequential Diagram of Getting a Thumbnail.

2.9 The Sequential Diagram of Operation: Get the Device Information and Get Setting Options

The API (**AMBA_GET_DEVICEINFO**) is used to retrieve the information which is related to the camera device, such as the model name, logo path, firmware version, API version, and camera application type. The user can use this API to get the API version to decide how to do correct operations because there may be some modifications when the API versions are different. For example, the streaming settings have been modified in the API version v2.8.0.

(Please refer to the document *AMBARELLA Application Note: Streaming Settings for Remote Control APIs for Versions later than v2p8p0 (General)* for more details)

The API **AMBA_GET_SETTING** is used to retrieve all types of setting value(s). If the command **AMBA_GET_ALL_CURRENT** SETTINGS retrieves a snapshot of all current values for all parameters, this command **AMBA_GET_SINGLE_SETTING_OPTIONS** retrieves all of the possible options for ONE SINGLE SETTING. This should enable the handheld application to display all possible options with, for instance, a pull down menu.

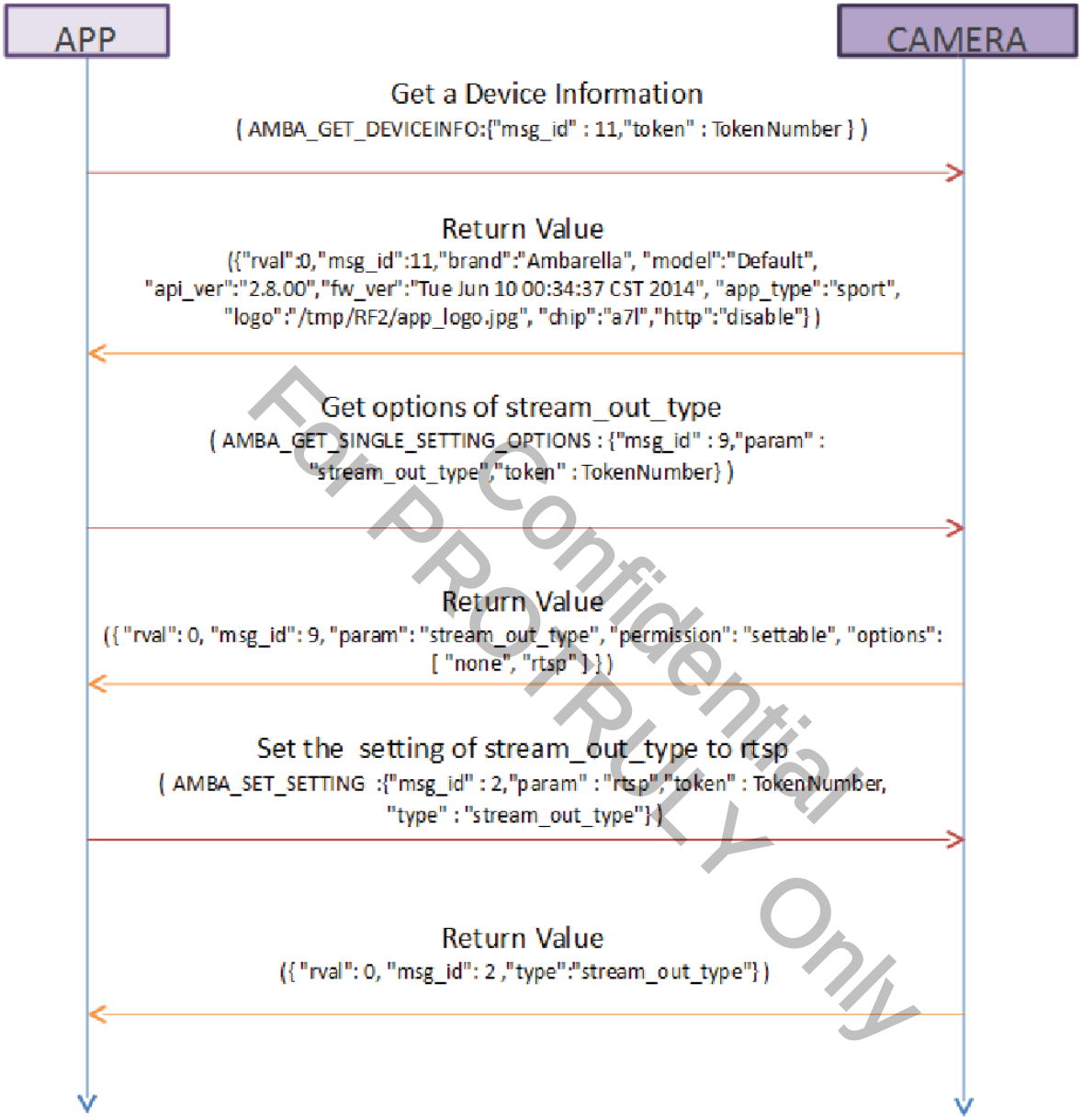

Figure 2-12. The Sequential Diagram of Getting a Device Information and Setting the Streaming to RTSP. (Please also refer to the section 2.2 about the limitations of setting streaming).

2.10 The Sequential Diagram of Operation: Upload File

The API (**AMBA_PUT_FILE**) is used to upload a file into the SD card in the camera. This command takes place in a two-step fashion. First, the camera begins receiving the specified file, assuming that (1) there is sufficient free space, and (2) the handheld has proper write permission at the destination directory. The return value, rval, only implies the successful commencement of reception.

The outcome of the **PUT_FILE** command will end in three possible outcomes (1) Success, (2) Out of Camera Storage Space, or (3) Interrupted mid-transmission and timed out. For outcome 1, upon the completion of file reception, as the second step the camera will send a completion notification **AMBA_NOTIFICATION > PUT_FILE_COMPLETE**. The smartphone should only infer a successful transfer when it receives this notification. For outcome 3, the param <bytes received> will denote the portion of the file successfully received and the offset argument can be used to continue **AMBA_PUT_FILE** from the last successful location.

(Please refer to the document *AMBARELLA A-Series Wireless Connectivity API: Remote Control for more details*)
Note: Send **AMBA_SET_CLNT_INFO** command to specify client information before **AMBA_PUT_FILE** starts.

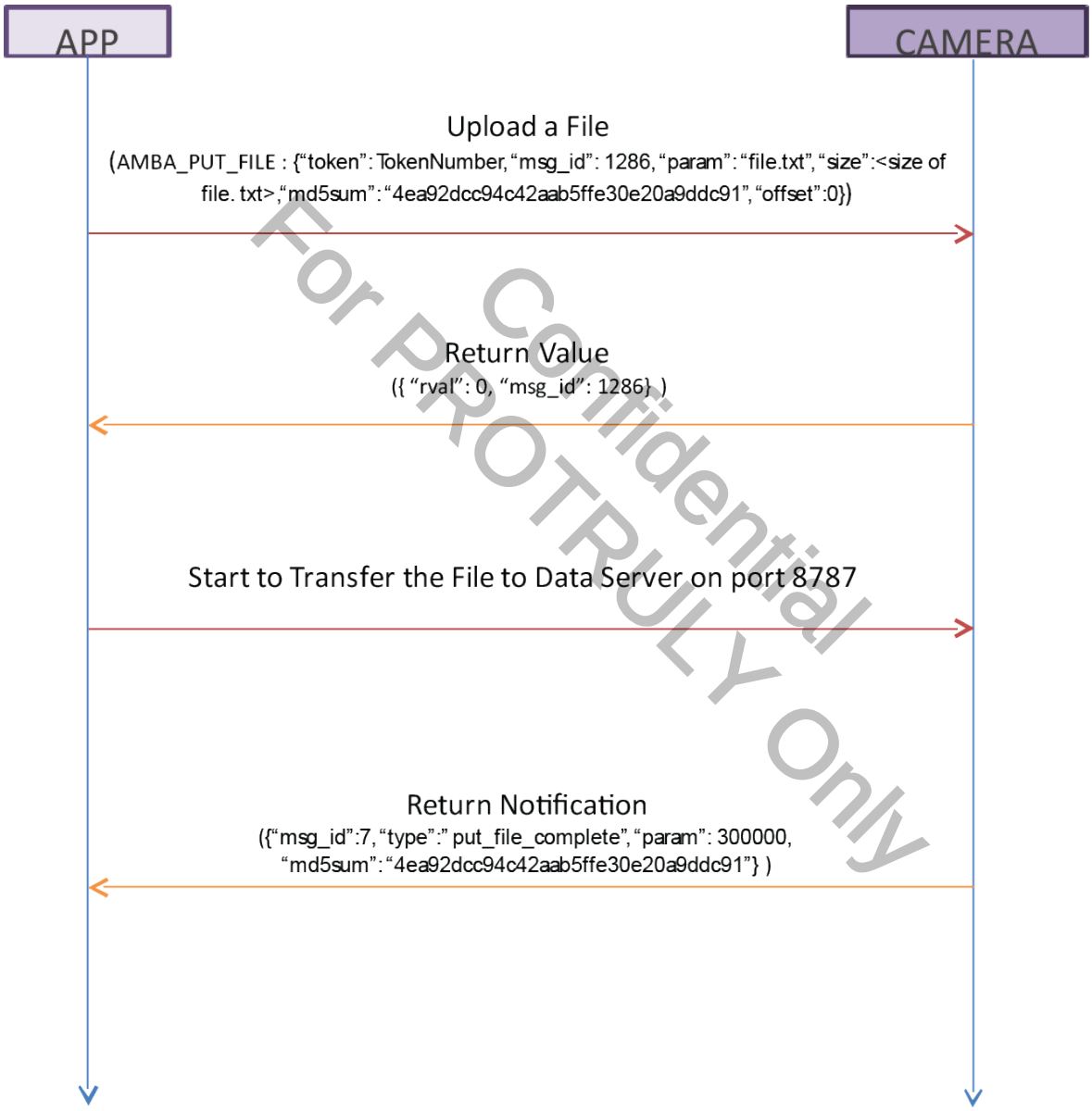

Figure 2-13. The Sequential Diagram of Uploading a File Successfully.

2.11 The Sequential Diagram of Operation: Burn Firmware

The API (**AMBA_BURNIN_FW**) programs a camera after an upgraded firmware (FW) image has been uploaded. This step represents the final step of the FW upgrade process.

Caution should be taken not to wait too long between the completion of the FW upgrade file upload and **AMBA_BURNIN_FW** as the battery level may decrease below operational levels. Before the **AMBA_BURNIN_FW** starts, the camera will check appropriate battery levels.

After completing the command, the camera will be automatically rebooted. If the FW upgrade is successful, the notification **AMBA_NOTIFICATION > FW_UPGRADE_COMPLETE** will be sent before the camera is rebooting.

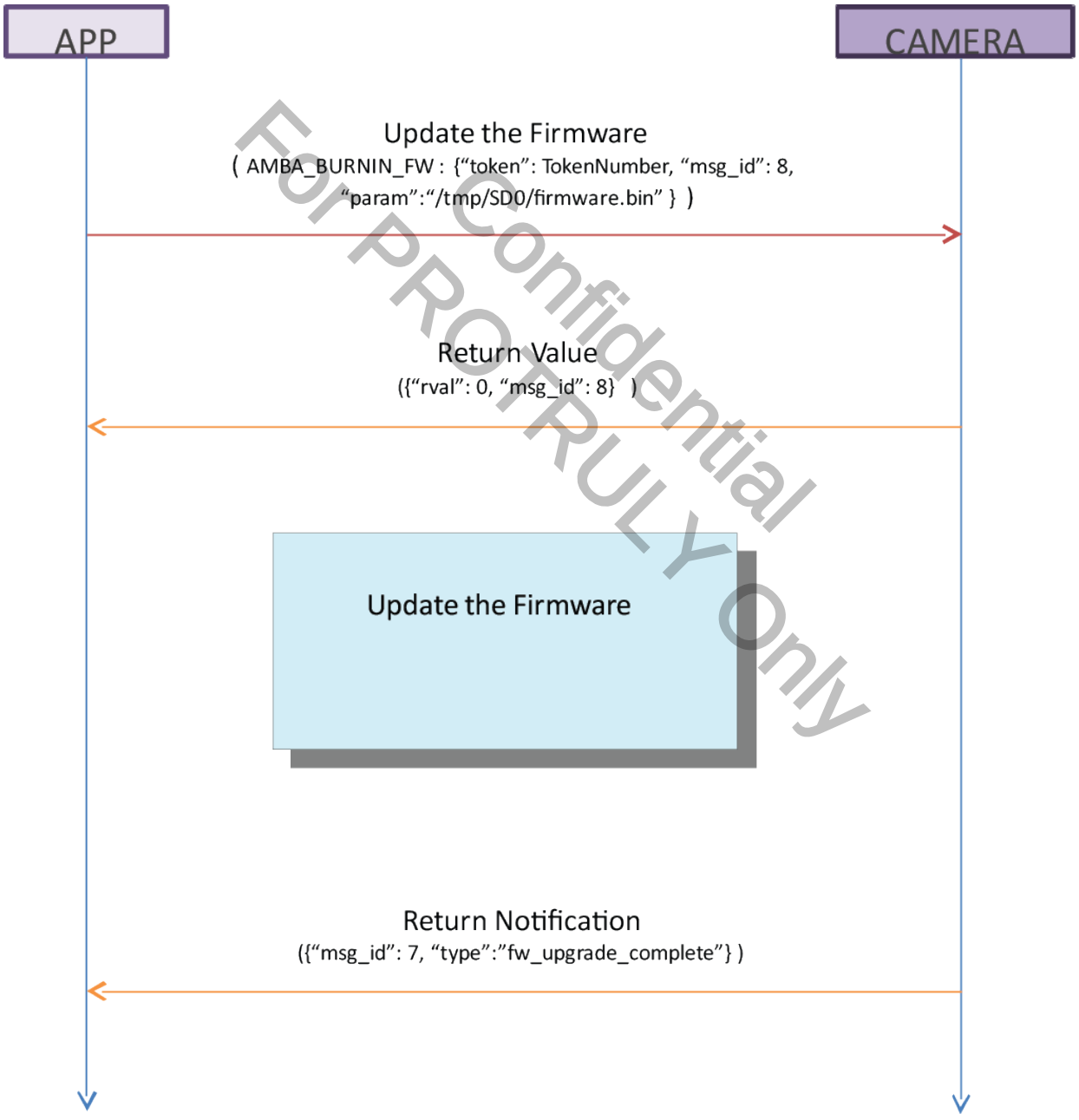

Figure 2-14. The Sequential Diagram of Upgrading the Firmware Successfully.

2.12 The Sequential Diagram of Operation: Enable Time-Lapse Video and Record

The client could use API **AMBA_SET_SETTING** to turn on/off Time Lapse video and **AMBA_RECORD_START** to trigger time-lapse video recording. If the param field of **timelapse_video** is specified as “off”, the **AMBA_RECORD_START** API will trigger normal video recording. Otherwise, the **AMBA_RECORD_START** API will trigger timelapse video record and use the setting specified in the param field as the period to capture each video frame.

(Please refer to the document *AMBARELLA A-Series Wireless Connectivity API: Remote Control* for more details)
NOTE: The Time-lapse is only supported in the Car-DV.

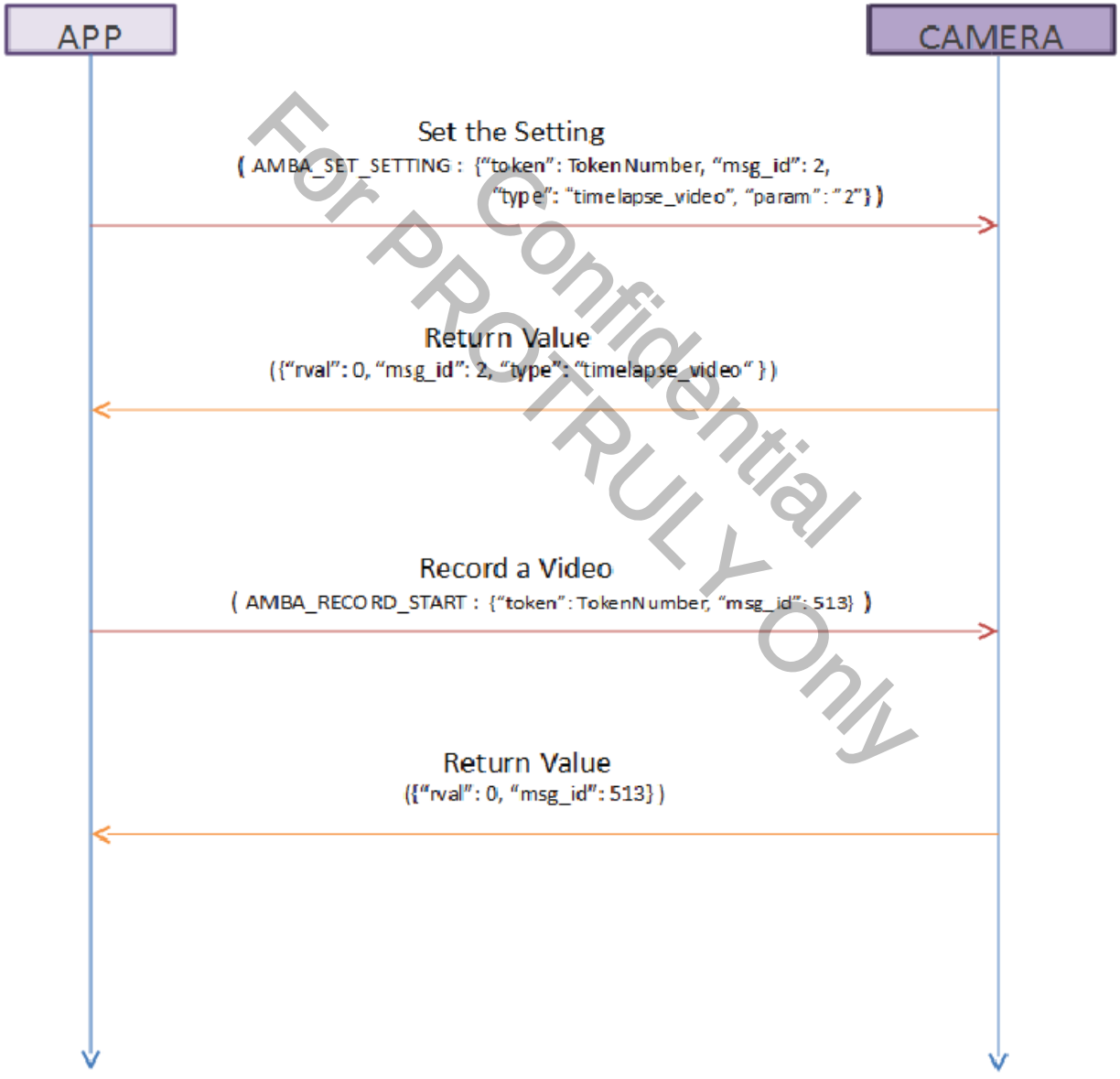

Figure 2-15. The Sequential Diagram of Turning on the Time-lapse Video and Recording.

2.13 The Sequential Diagram of Operation: Change the Capture Mode and Take Photos

The API **AMBA_SET_SETTING** -> **capture_mode** is used to specify the photo capture mode.

The API (**AMBA_TAKE_PHOTO**) is used to capture a still image, begin a precise continuous capture or start to perform burst continuous capture depending on the value of **AMBA_GET_SETTING** -> **capture_mode**. If there is no **capture_mode** available in the **AMBA_GET_ALL_CURRENT_SETTING**, the camera only supports precise capture (single photo). Note that when capture_mode= “precise_cont” or “burst_cont” (continuous capture), the hand-held device is obligated to send **AMBA_CONTINUE_CAPTURE_STOP** to stop the operation of continuous capture.

Different photo capture modes may have different image size options. This leads to the dependency between modes and image sizes. Please retrieve options for photo size after changing the **capture_mode**.

- Note that some of the camera operations can only be enabled at some specific **capture_mode**. For example, the user must set **capture_mode** as “precise cont” so that time-lapse photo setting can be settable (Otherwise, this setting will be read-only). Furthermore, when the time-lapse photo is enabled, it will make **capture_mode** locked at “precise cont”. Please disable time-lapse photo if the user wants to change **capture_mode**.

(Please refer to the document *AMBARELLA A-Series Wireless Connectivity API: Remote Control* for more details)

NOTE: This flow is only supported in the Sport-DV.

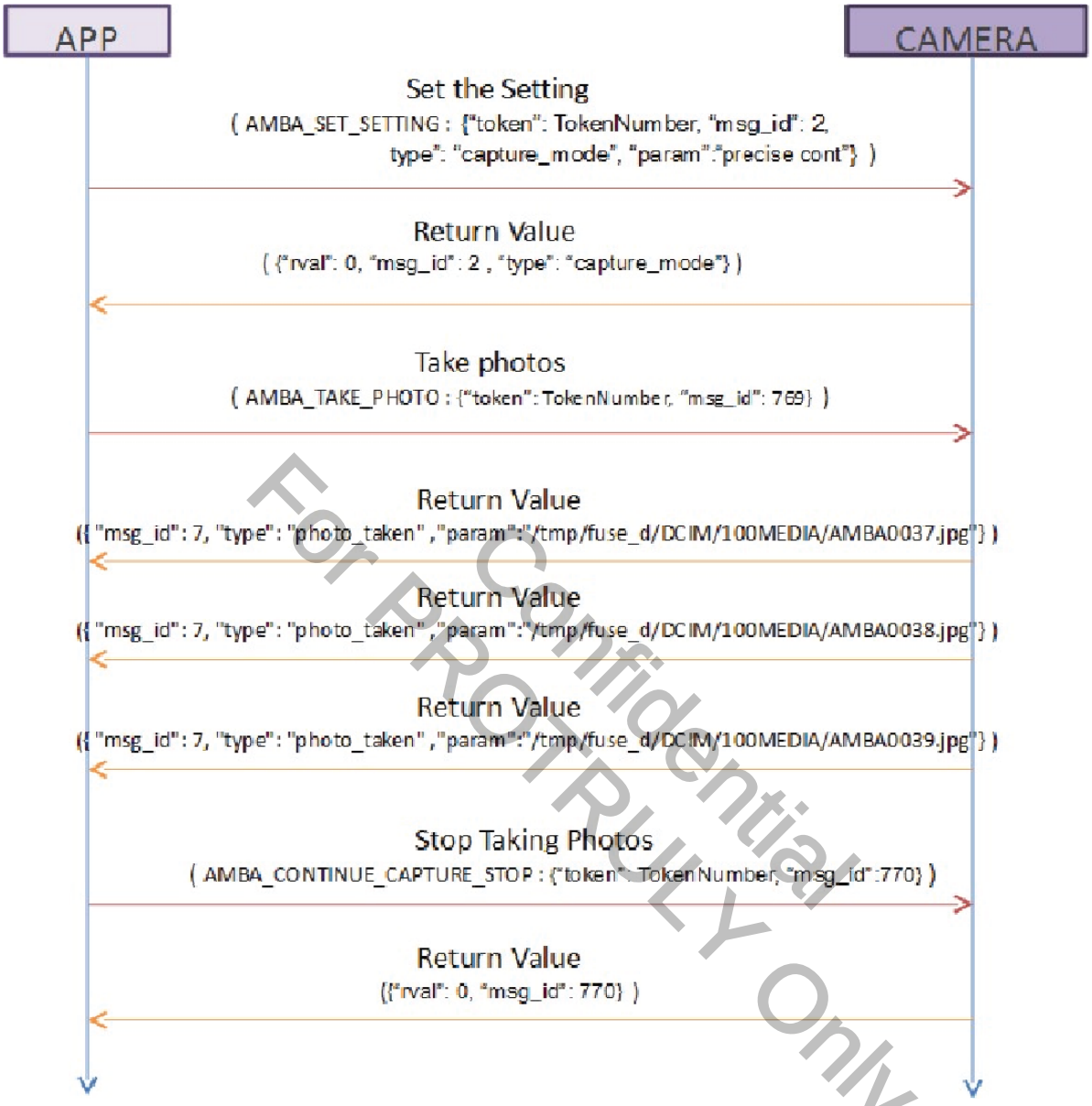

Figure 2-16. The Sequential Diagram of Taking Photos Continuously in the Mode "precise cont".

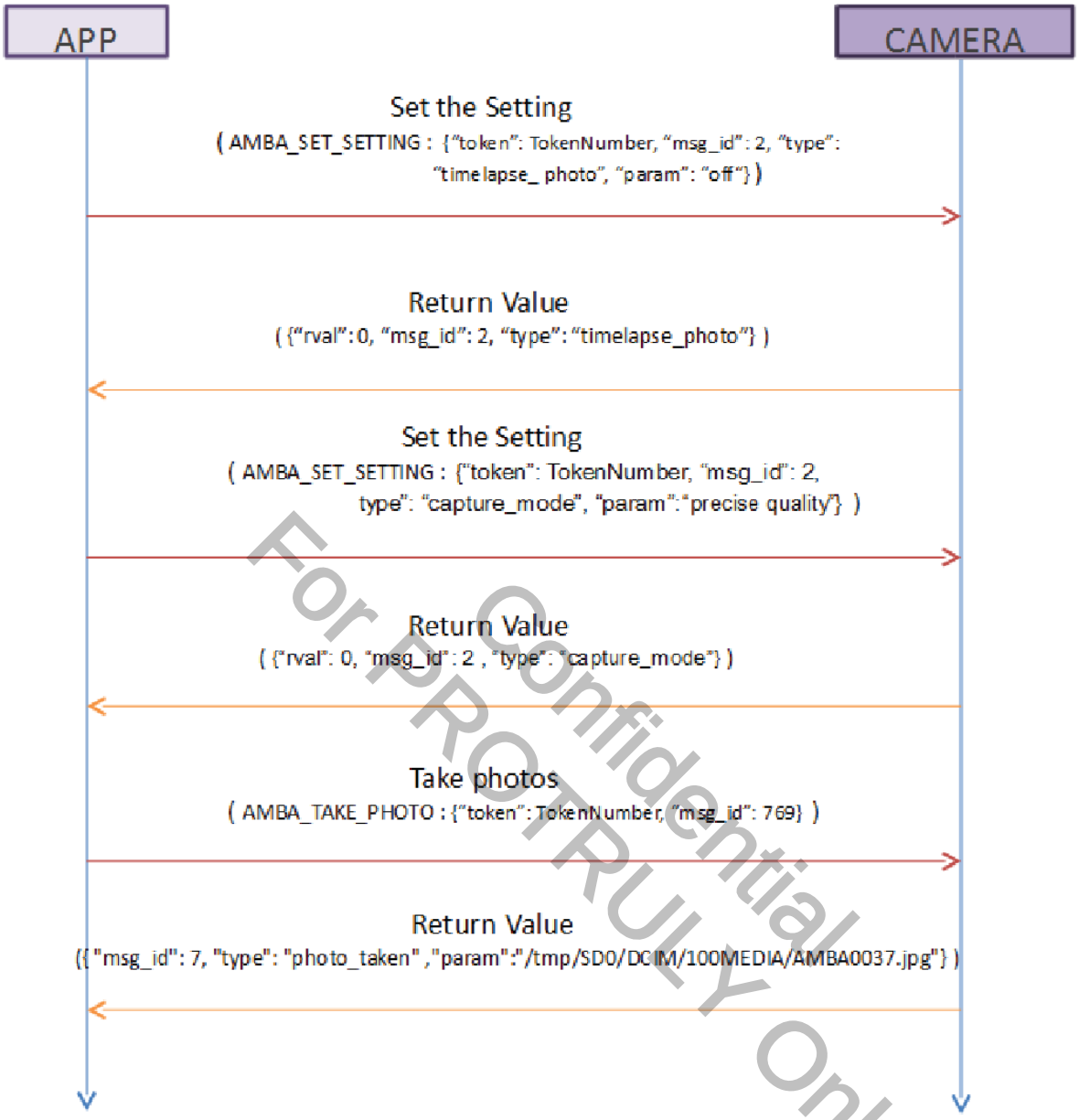

Figure 2-17. The Sequential Diagram of Changing the Capture Mode and Taking the Photo when the Time-lapse Photo is Enabled.

2.14 The Sequential Diagram of Operation: Switch the Mode of Camera to Standby

The API (**AMBA_POWER_MANAGE**) is used to either switch the camera operating mode to standby mode or to power-off the camera device. After executing this command, the camera will be switched to the specified mode or will be powered-off. If the operation is successful, the notification **AMBA_NOTIFICATION** > **POWER_MODE_CHANGE** will be sent to the handheld.

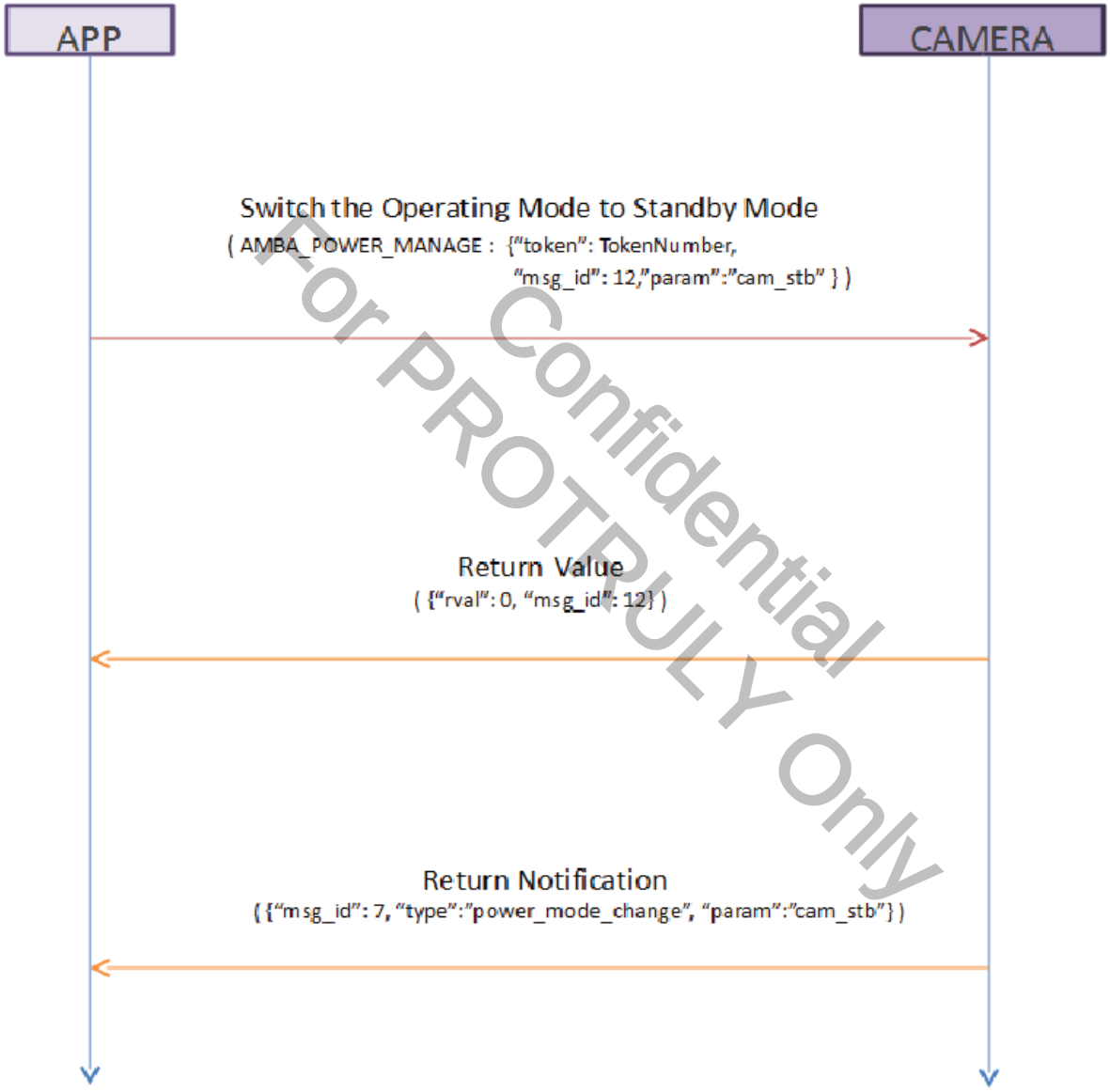

Figure 2-18. The Sequential Diagram of Switching the Operating Mode to Standby Mode.

2.15 The Sequential Diagram of Operation: Wake the Camera Up

This section describes the flow of waking the camera up. If the current mode of the camera is at standby, the application can send he UDP packet with “amba discovery” message to the camera on port 7877 to wake it up not only by using WIFI but also Bluetooth.

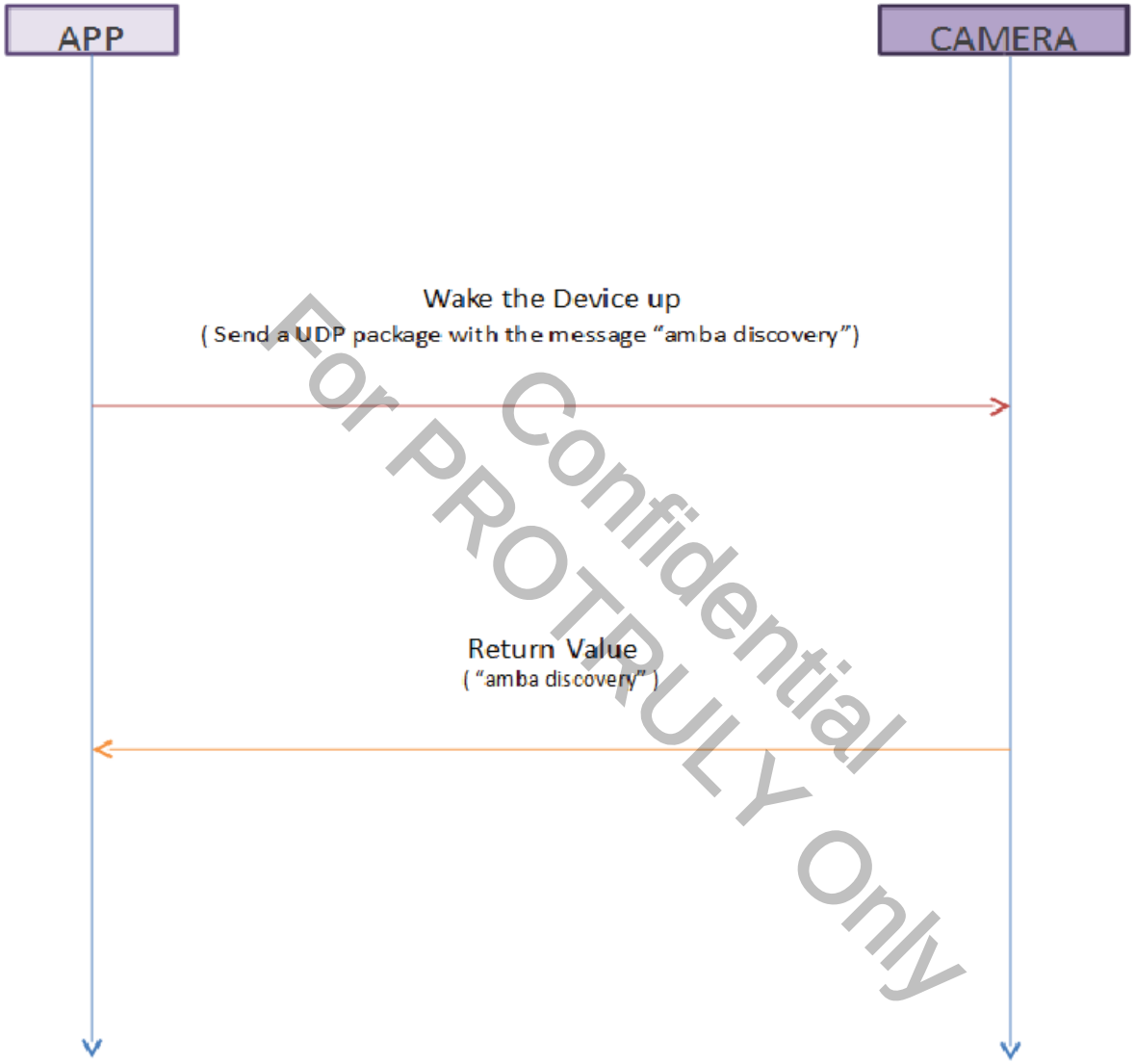

Figure 2-19. The Sequential Diagram of Waking the Camera Up by Using WIFI.

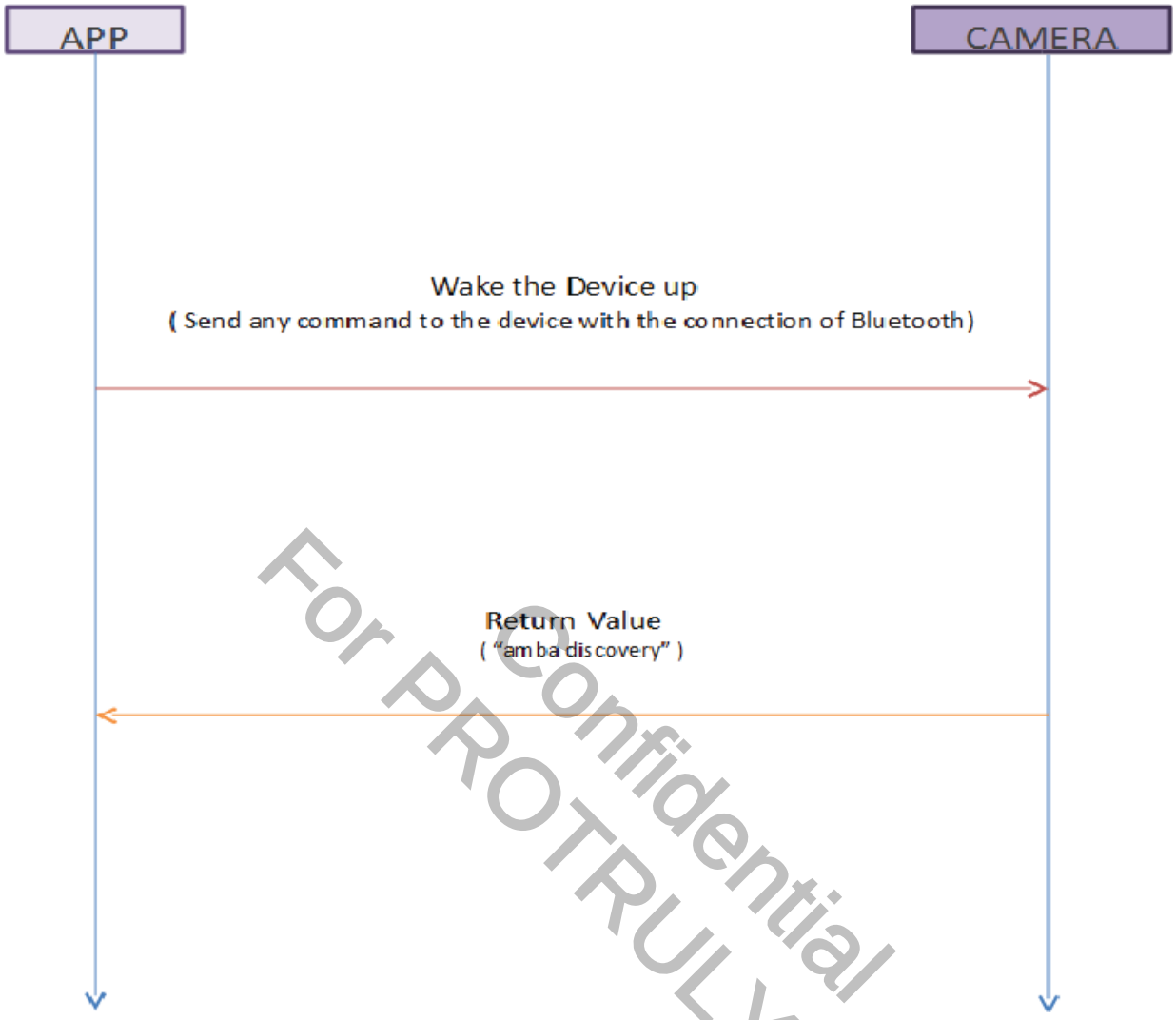

Figure 2-20. The Sequential Diagram of Waking the Camera Up by Using Bluetooth.

3 How to Design the Client Application and Notice

3.1 How to Design the Client Application and Notice: Overview

This chapter provides the explanation for designing Client application and notice.

3.2 How to Design the Client Application and Notice: Establish Connection

Before the client sends any request command with JavaScript Object Notation (JSON), it should connect to command server on port 7878 for WiFi, port 20 for Bluetooth (BT), or GATT for Bluetooth Low Energy (BLE) (use the channel, 11111111-616d-6261-5f69-645f62617365 to send data to the camera, and use the channel 33333333-616d-6261-5f69-645f62617365 to receive data from the camera). After the session starts, it also should connect to the data server on port 8787 for WiFi, or port 10 for BT to transfer data between the client and the camera. Therefore, there are two connections between the client and camera, one is used for command service, and the other one is used for data service. The client will keep two connections until it wants to disconnect, and the camera will not issue close connection with the client actively.

3.3 How to Design the Client Application and Notice: Get Valid Token

The camera has a valid token management mechanism. Therefore, the client should check whether it can get a valid token or not. If the client fails to get a valid token and start session, the client should break this connection with the camera and check what error has occurred. When the connection is broken, it means that the valid token has been acquired by other client. That is to say, the camera is controlled by one client at the same time. It is not possible to control the camera through multiple clients at the same time.

3.4 How to Design the Client Application and Notice: Response and Notification

The client must check whether it can get the return value of zero or not after it sends out the request command. If the client gets a non-zero return value, it should perform appropriate error handling. However, the client also should handle notification which the camera sends out. Notification is not necessary for each request command which the client sends out, but the response is necessary for each request command which the client sends out.

3.5 How to Design the Client Application and Notice: Transfer Data

Before the client starts to transfer data, it must setup active client information which is the IP address for WiFi and Media Access Control (Mac) address for BT. If the client setup has the incorrect information, the camera cannot perform any data transmission operation. Therefore, the client must guarantee that the client information is correct before sending it to the camera. After the data transmission is done, the client will receive notification with md5 checksum to check whether the data is complete or not.

Appendix 1 Additional Resources

Please contact an Ambarella representative for digital copies.

Appendix 2 Important Notice

All Ambarella design specifications, datasheets, drawings, files, and other documents (together and separately, “materials”) are provided on an “*as is*” basis, and Ambarella makes no warranties, expressed, implied, statutory, or otherwise with respect to the materials, and expressly disclaims all implied warranties of noninfringement, merchantability, and fitness for a particular purpose. The information contained herein is believed to be accurate and reliable. However, Ambarella assumes no responsibility for the consequences of use of such information.

Ambarella Incorporated reserves the right to correct, modify, enhance, improve, and otherwise change its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

All products are sold subject to Ambarella’s terms and conditions of sale supplied at the time of order acknowledgment. Ambarella warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with its standard warranty. Testing and other quality control techniques are used to the extent Ambarella deems necessary to support this warranty.

Ambarella assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using Ambarella components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

Ambarella does not warrant or represent that any license, either expressed or implied, is granted under any Ambarella patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which Ambarella products or services are used. Information published by Ambarella regarding third-party products or services does not constitute a license from Ambarella to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from Ambarella under the patents or other intellectual property of Ambarella.

Reproduction of information from Ambarella documents is not permissible without prior approval from Ambarella.

Ambarella products are not authorized for use in safety-critical applications (such as life support) where a failure of the product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Customers acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of Ambarella products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by Ambarella. Further, Customers must fully indemnify Ambarella and its representatives against any damages arising out of the use of Ambarella products in such safety-critical applications.

Ambarella products are neither designed nor intended for use in automotive and military/aerospace applications or environments. Customers acknowledge and agree that any such use of Ambarella products is solely at the Customer’s risk, and they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

Appendix 3 Revision History

NOTE: Page numbers for previous drafts may differ from page numbers in the current version.

Version	Date	Comments
0.1	12 February 2015	Preliminary version

Table A3-1. Revision History.