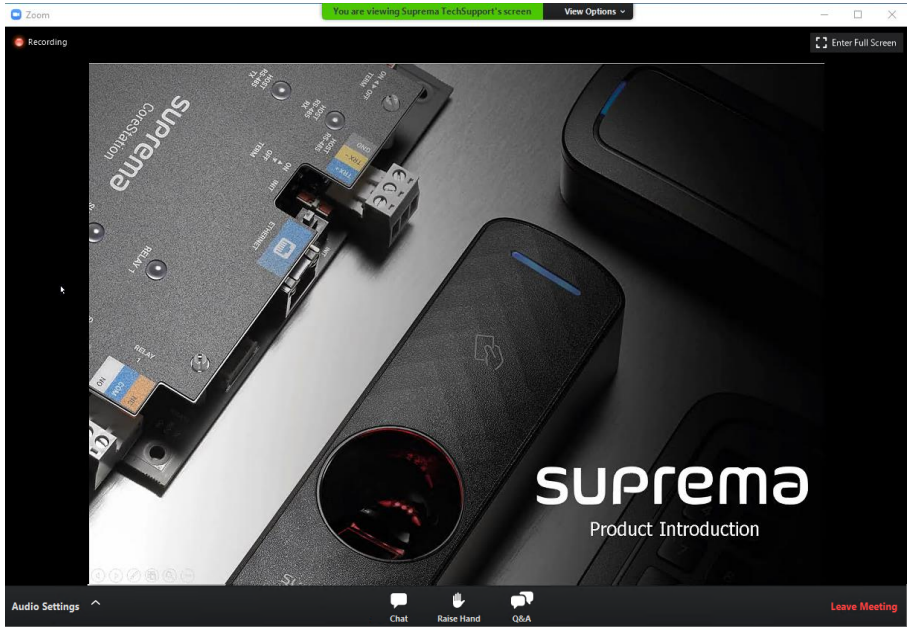
Suprema Webinar
G-SDK New Features v1.2

Speaker
Simon Lee
Suprema NA Inc.

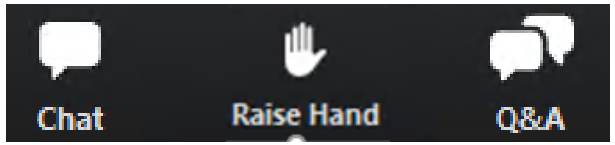
Date: October 21, 2020
Time: 11:30 AM (GMT-4) Toronto, Canada

Date: October 23, 2020
Time: 1:30 PM (GMT+9) Seoul, Korea

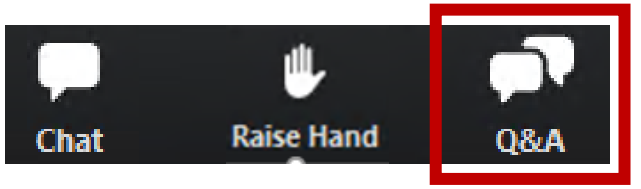
How can you ask questions during a webinar?



Please mouse to the bottom of your screen, the 3 icons will appear.



1) [Q&A] Box for leaving your question



✓ During the webinar, you can leave your questions on the **Q&A box anytime**. Suprema agents will answer in real time.

Contents

1. G-SDK Overview
2. V1.1 New Features
3. V1.2 New Features
4. Roadmap
5. Q&A

G-SDK Overview

Design Goals

- Scalable and extensible
  - Handle thousands of devices
  - Easy to maintain and customize
- Multi-language support
  - Language-neutral IDL
  - Native client libraries
- Well-defined API
  - Easy to understand and use
- Mobile/cloud ready
  - Easily deployable on Cloud
  - Accessible from mobile devices directly

Target Customers

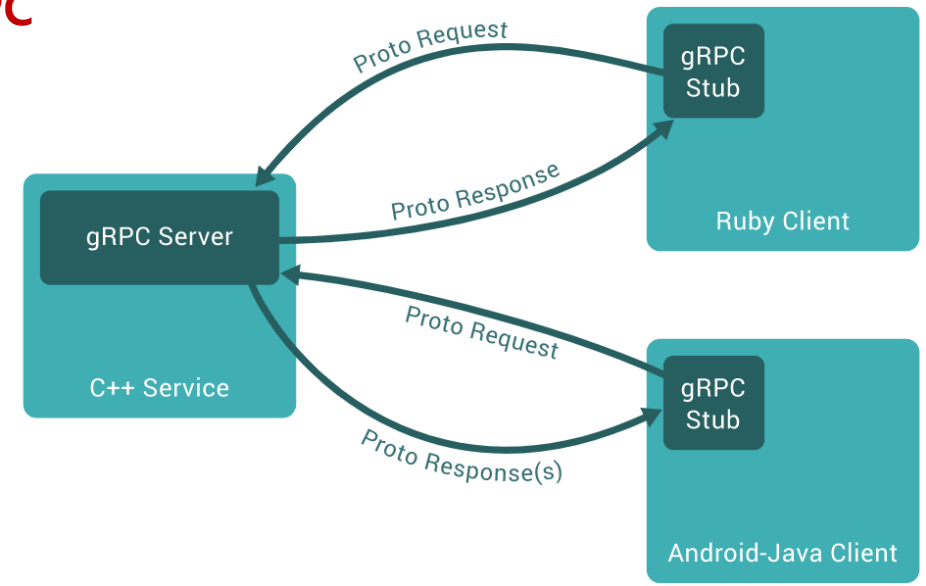
- Existing
  - Users of BioStar API
  - Users dissatisfied with the Device SDK
    - Scalability and stability
    - Language support
- New
  - Developers with other languages than C/C++/C#
  - Mobile & cloud applications with BioStar devices
  - T&A applications

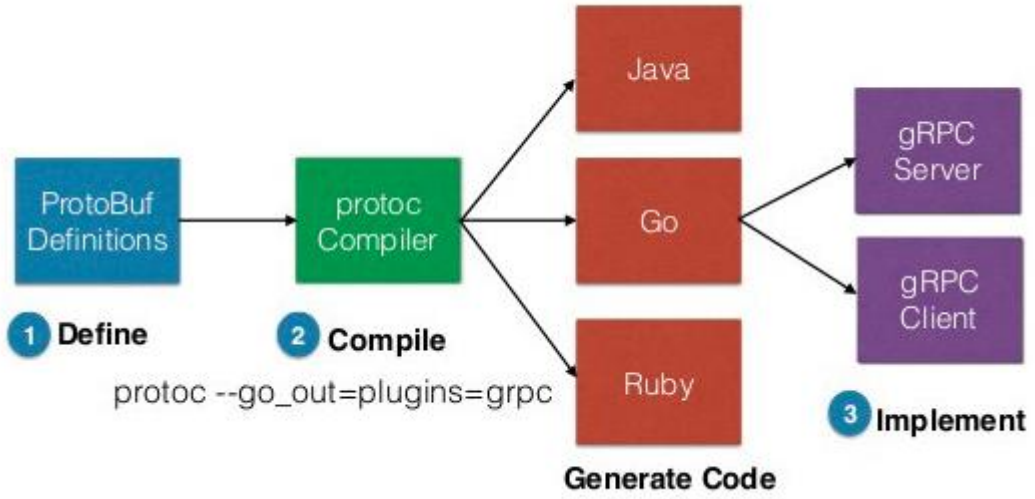
G-SDK Overview: Some Technical Details

Why go?

- Optimized for handling lots of small tasks
  - goroutine/channel
  - Much simpler architecture compared to the thread-pool model
- Easy to use
  - Garbage collection, rich standard libraries, etc.
- Hard to abuse
  - Strict naming convention, dependency checking
- Easy to deploy
  - Static linking by default
  - One self-sufficient binary

gRPC





G-SDK Overview: vs. Device SDK

	Device SDK	G-SDK
Deployment	• Shared library	• Client libraries • Device Gateway • (Master Gateway)
OS	• Windows • x86 Linux	• Windows • x86/Arm Linux • MacOS
Supported Language	• C++ • C# example	• As of V1.1, Java, C#, Python, Go, Node.js, and C++ are supported
Max. Devices	• 1,000	• 1, 000 (Device Gateway) • 100, 000 (Master Gateway)

G-SDK Overview: Job Code Example

Device SDK

- Connection
  - *BS2_ConnectDeviceViaIP*
  - Connection callback
- User
  - *BS2_EnrollUser/BS2_GetUserInfos*
- Event
  - *BS2_GetLogBlob*

G-SDK

- Connection
  - *connect.AddAsyncConnection*
  - One call for multiple devices
  - Handled automatically by the device gateway
- User
  - *user.EnrollMulti*
  - *tna.SetJobCodeMulti*
- Event
  - *tna.GetJobCodeLog*

G-SDK Overview: Job Code Example (cont.)

Device SDK

```
typedef struct {
  uint16_t eventMask;
  BS2_EVENT_ID id;
  BS2EventExtInfo info;
  union
  {
    BS2_USER_ID userID;
    uint8_t cardID[BS2_CARD_DATA_SIZE];
    BS2_DOOR_ID doorID;
    BS2_ZONE_ID zoneID;
    BS2EventExtIoDevice ioDevice;
  };
  BS2_TNA_KEY tnaKey;
  BS2_JOB_CODE jobCode;
  uint16_t imageSize;
  uint8_t image[BS2_EVENT_MAX_IMAGE_SIZE];
  uint8_t reserved;
}BS2EventBlob;
```

G-SDK: protobuf

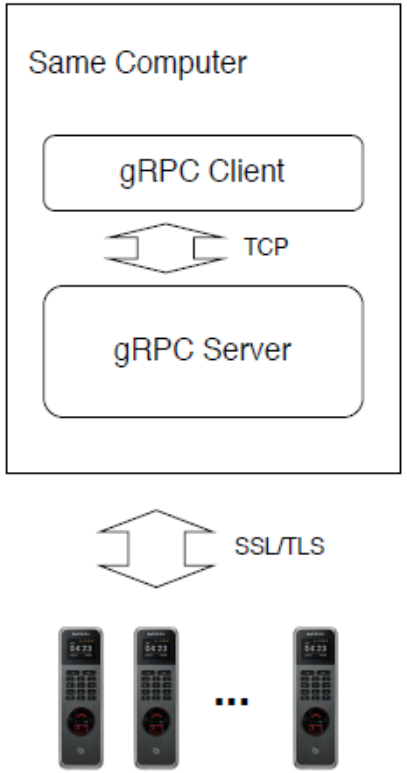
```
message EventLog {
  uint32 ID;
  uint32 timestamp;
  uint32 deviceID;
  string userID;
  uint32 entityID;
  uint32 eventCode;
  uint32 subCode;
  tna.Key TNAKey;
  bool hasImage;
}
message JobCodeLog {
  uint32 ID;
  uint32 timestamp;
  uint32 deviceID;
  string userID;
  uint32 eventCode;
  uint32 subCode;
  uint32 jobCode;
}
```

G-SDK: Client Languages

```
public sealed partial class JobCodeLog : pb::IMessage<Job
CodeLog> { // C#


public  final class JobCodeLog extends

    com.google.protobuf.GeneratedMessageV3 implements

    JobCodeLogOrBuilder {// Java


type JobCodeLog struct { // go

  ID                    uint32    `protobuf:"varint,1,opt,n


proto.tna.JobCodeLog = function(opt_data) { // nodejs
```

G-SDK Overview: Deployment

Same as DLL

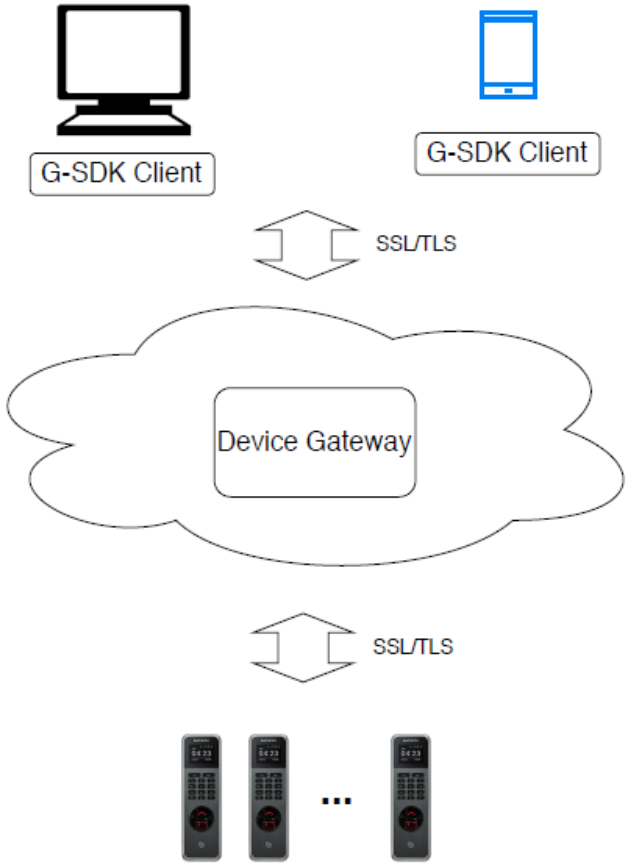


How to deploy?

- Install both the server and the client S at the same machine

- Virtually identical to a DLL

Gateway on the Cloud



How to deploy?

- You don't have to install both the gateway and client at the same machine.

- A gateway can handle multiple clients at the same time.

- By installing the gateway a cloud, you can easily develop mobile and web applications.

Issues

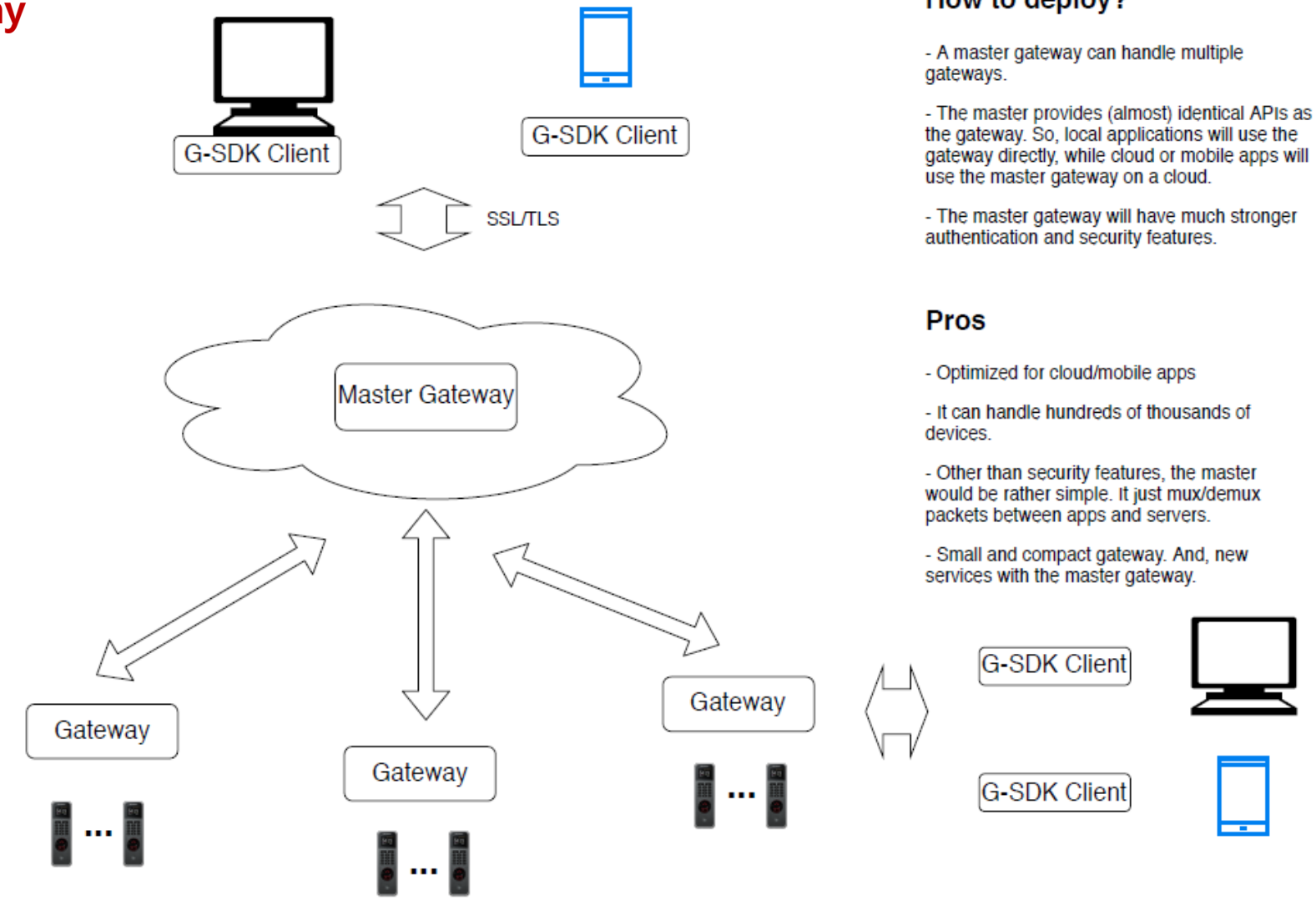
- Some clients may not want their devices are connected directly to the gateway in a cloud for security reasons.

- To deploy the gateway on a cloud, it should have many new features such as strong authentication, which would make its architecture much more complex.

- The gateway will be able to handle thousands of devices. However, how about tens of thousands of devices? And, even with smaller number of devices, how to support scattered installations?

V1.1 New Features

Master Gateway



How to deploy?

- A master gateway can handle multiple gateways.
- The master provides (almost) identical APIs as the gateway. So, local applications will use the gateway directly, while cloud or mobile apps will use the master gateway on a cloud.
- The master gateway will have much stronger authentication and security features.

Pros

- Optimized for cloud/mobile apps
- It can handle hundreds of thousands of devices.
- Other than security features, the master would be rather simple. It just mux/demux packets between apps and servers.
- Small and compact gateway. And, new services with the master gateway.

V1.2 New Features: Advanced Access Control

Zones

- APB, Timed APB, Fire Alarm, Intrusion Alarm, Scheduled Lock, Interlock

Zone API

You can configure several types of zones for high-level functions such as anti passback and intrusion alarm. G-SDK supports only local zones, where all devices should be within a RS485 network.

Status

```
message ZoneStatus {
  uint32 zoneID;
  Status status;
  bool disabled;
}
```

zoneID

The ID of the zone.

Zone
Status
Anti passback zone
Timed anti passback zone
Intrusion alarm zone
Fire alarm zone
Scheduled lock zone
Interlock zone

Lift

- OM-120

Lift API

OM-120 is an output extension module which controls up to 12 output relays. It can be used as an elevator controller. Maximum 31 units can be connected to a master device via RS485. Refer to the article for examples.

Information

```
message LiftInfo {
  uint32 liftID;
  string name;

  repeated uint32 deviceIDs;

  uint32 activateTimeout;
```

Lift
Information
GetList
GetStatus
Management
Add
Delete
DeleteAll
Activate/Deactivate
Activate
Deactivate
Release
SetAlarm

V1.2 New Features: FaceStation F2

Visual + IR

- New data format
- Same API

```
enum FaceFlag {
  BS2_FACE_FLAG_NONE = 0x00;
  BS2_FACE_FLAG_F2 = 0x100;
}

message FaceData {
  int32 index;
  uint32 flag;
  repeated bytes templates;
  bytes imageData;

  // Only for FaceStation F2
  repeated bytes irTemplates;
  bytes irImageData;
}
```

Multi-modal

- F2FP
- New authentication modes

```
// The below modes are only for FaceStation F2
AUTH_EXT_MODE_FACE_ONLY    = 11;
AUTH_EXT_MODE_FACE_FINGERPRINT = 12;
AUTH_EXT_MODE_FACE_PIN = 13;
AUTH_EXT_MODE_FACE_FINGERPRINT_OR_PIN = 14;
AUTH_EXT_MODE_FACE_FINGERPRINT_PIN = 15;

AUTH_EXT_MODE_FINGERPRINT_ONLY = 16;
AUTH_EXT_MODE_FINGERPRINT_FACE = 17;
AUTH_EXT_MODE_FINGERPRINT_PIN = 18;
AUTH_EXT_MODE_FINGERPRINT_FACE_OR_PIN = 19;
AUTH_EXT_MODE_FINGERPRINT_FACE_PIN = 20;

AUTH_EXT_MODE_CARD_ONLY = 21;
AUTH_EXT_MODE_CARD_FACE = 22;
AUTH_EXT_MODE_CARD_FINGERPRINT = 23;
AUTH_EXT_MODE_CARD_PIN = 24;
AUTH_EXT_MODE_CARD_FACE_OR_FINGERPRINT = 25;
AUTH_EXT_MODE_CARD_FACE_OR_PIN = 26;
```

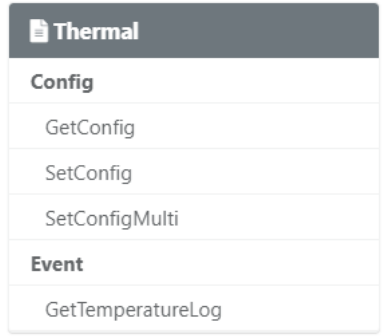
V1.2 New Features: Thermal Camera

Supported Models

- FaceStation 2
- FaceStation F2
  - Checking masks

Thermal Camera API

Suprema Thermal Camera can be used in combination with face recognition terminals to detect users with elevated skin temperature. With ThermalConfig, you can specify the options related to the camera. You can also read log records with temperature information.



Config

```
message ThermalConfig {
  CheckMode checkMode;
  CheckOrder checkOrder;
  TemperatureFormat temperatureFormat;
  uint32 temperatureThreshold;

  bool auditTemperature;
  bool useRejectSound;
  bool useOverlapThermal;

  ThermalCamera camera;

  // Only for FaceStation F2
  CheckMode maskCheckMode;
  MaskDetectionLevel maskDetectionLevel;
}
```

V1.2 New Features: Enhanced DB for Master Gateway

Connection API

- Synchronous
  - Simplest
  - One device at a time
- Asynchronous
  - Handled by the gateway in the background
  - Multiple devices
  - Have to reassign devices whenever the gateway restarts or reconnects
- Accept filter
  - Device-to-server mode

Asynchronous Connection DB

- Connection information is stored into the database of the master gateway
- Reconnections of gateways are handled by the master gateway automatically
- ConnectMaster API
  - AddAsyncConnectionDB
  - DeleteAsyncConnectionDB
  - SetAcceptFilterDB
  - GetAcceptFilterDB

V1.2 New Features: New Examples

User Example

- Enroll cards/fingerprint/faces
- Change authentication modes
- Read log records with filters
- Real-time event monitoring

Thermal/T&A Example

- Change options
- Read temperature/T&A logs
- Real-time event monitoring

```
String newUserID = String.format("%d", Instant.now().getEpochSecond());
List<UserInfo> newUsers = new ArrayList<UserInfo>();
UserHdr hdr = UserHdr.newBuilder().setID(newUserID).build();
UserSetting setting;

if(deviceType == Type.FACESTATION_F2 || deviceType == Type.FACESTATION_F2_FP) {
  setting = UserSetting.newBuilder()
              .setCardAuthExtMode(AuthMode.AUTH_EXT_MODE_CARD_ONLY_VALUE)
              .setFingerAuthExtMode(AuthMode.AUTH_EXT_MODE_FINGERPRINT_ONLY_VALUE)
              .setFaceAuthExtMode(AuthMode.AUTH_EXT_MODE_FACE_ONLY_VALUE)
              .build();
} else {
  setting = UserSetting.newBuilder()
              .setCardAuthMode(AuthMode.AUTH_MODE_CARD_ONLY_VALUE)
              .setBiometricAuthMode(AuthMode.AUTH_MODE_BIOMETRIC_ONLY_VALUE)
              .build();
}

newUsers.add(UserInfo.newBuilder().setHdr(hdr).setSetting(setting).build());
userSvc.enroll(deviceID, newUsers);
```

Roadmap: More Examples

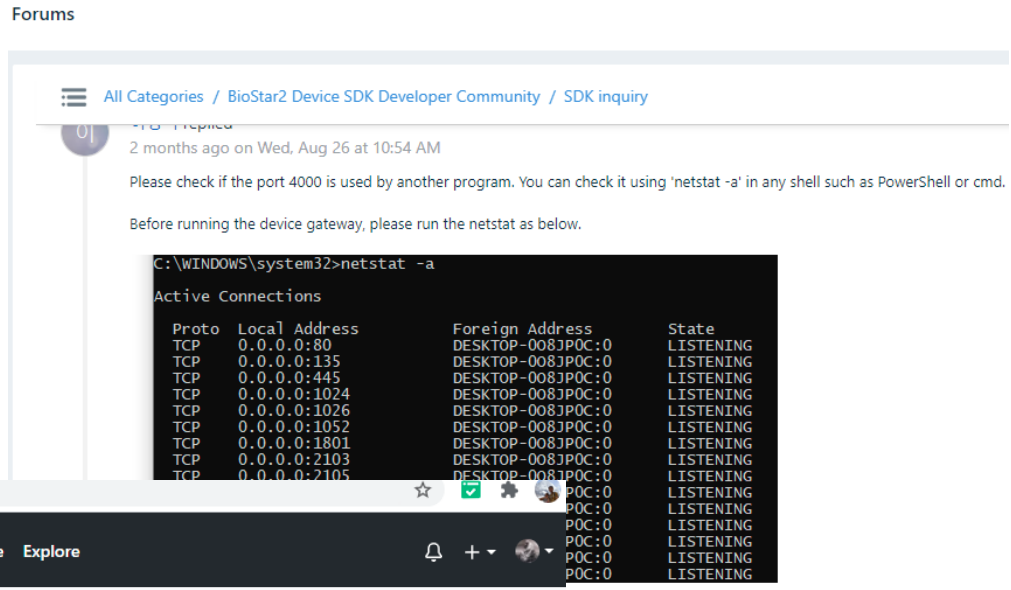
APIs

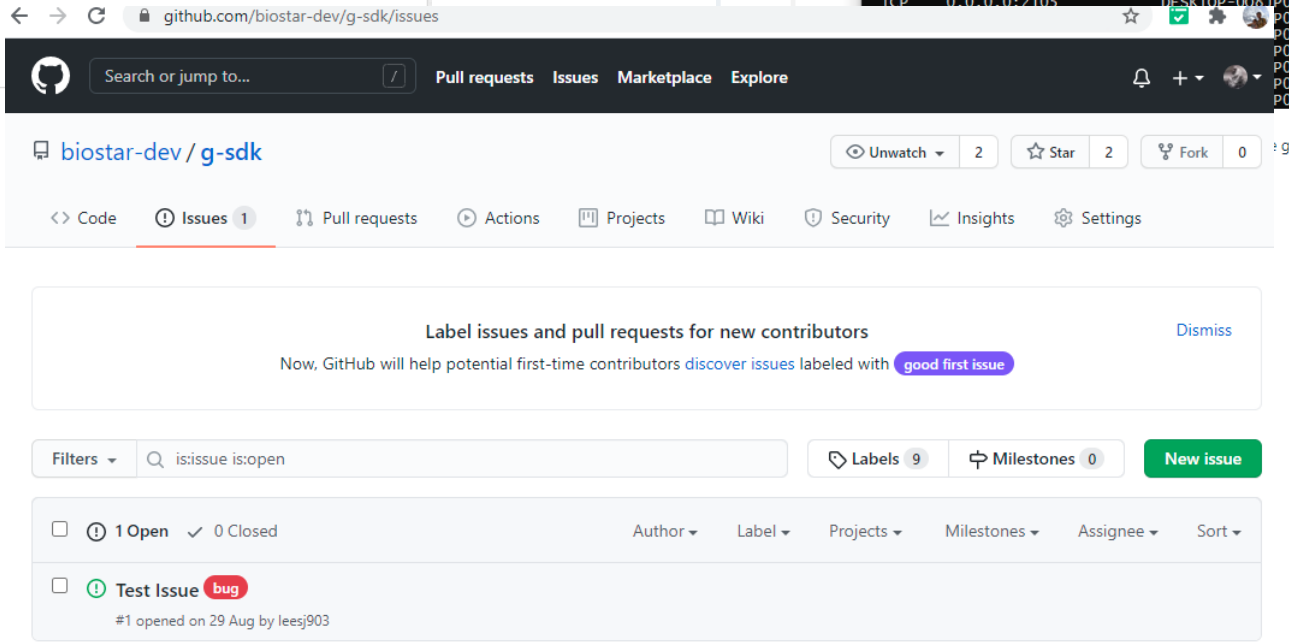
- V1.2
  - Quick start guide
  - Connect/ConnectMaster API
  - User/T&A/Thermal API
- V1.3 and later
  - Examples for major APIs such as Access, Door, RS485, Schedule, etc.
  - gRPC client debugging

Mobile/Cloud

- Master gateway
  - Management of multiple gateways/devices
  - User synchronization
- Mobile clients
  - iOS (Swift)
  - Android

Roadmap: Developer Support





Q&A

suprema
SECURITY & BIOMETRICS

20 Years of Innovation