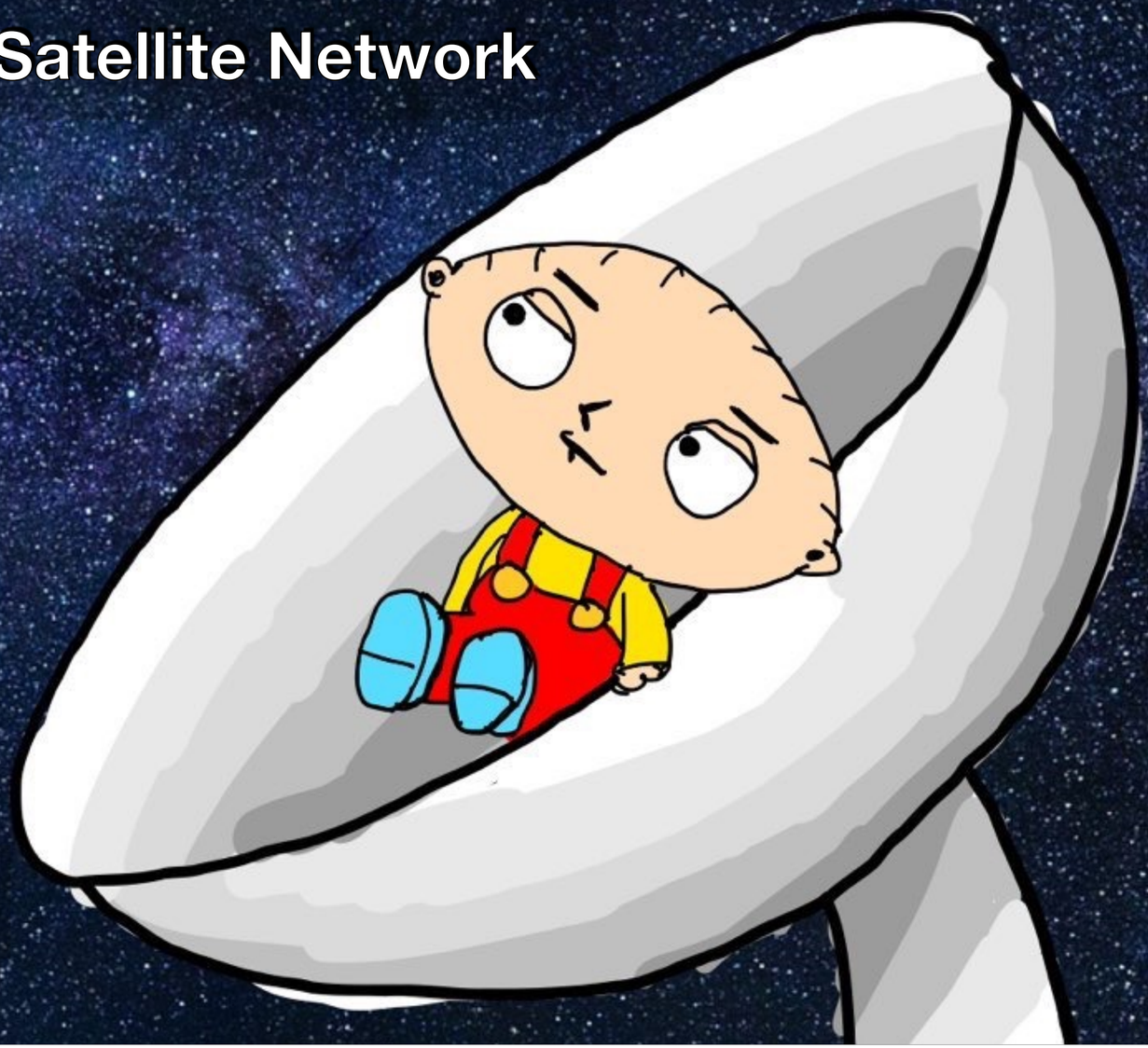
Lois... Ma... Mommy...

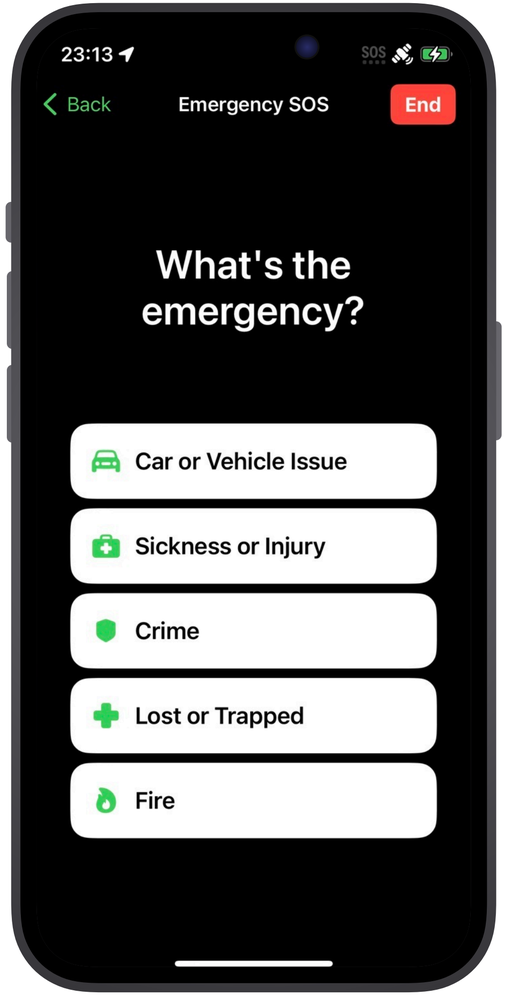
Stewie Talking to Apple's Satellite Network



Alexander Heinrich & Jiska Classen


ARA
Flugrettung
RK 2
OE-XAR
RRZT
ROCKSHAKE
ARA
Flugrettung

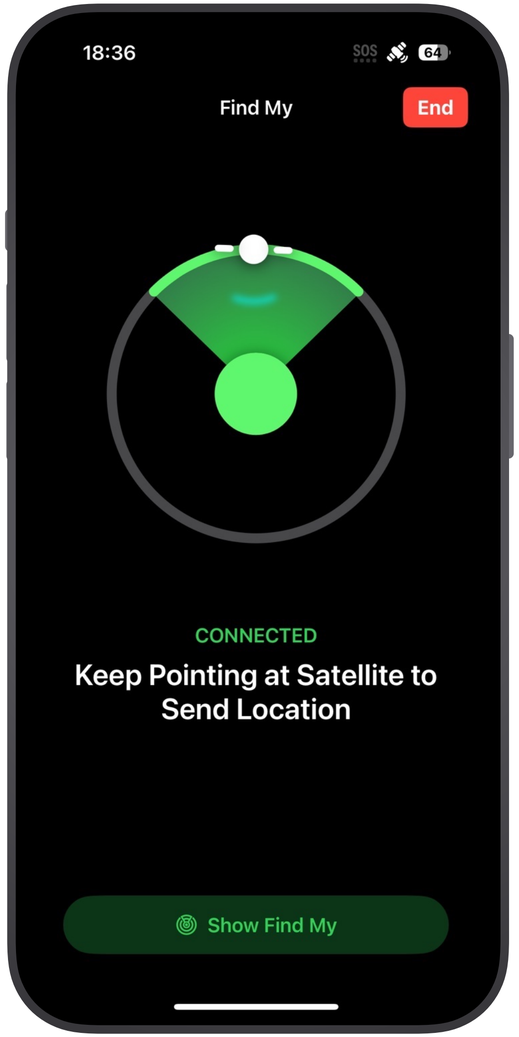
Emergency SOS

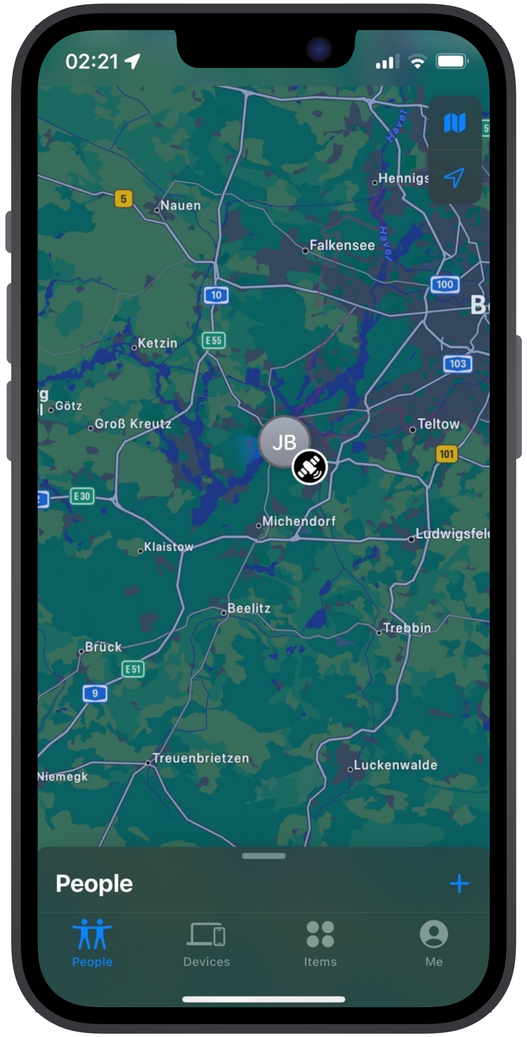


Requirements

- No Wi-Fi or cellular coverage
- Failed emergency call
- SOS button appears in phone menu

Find My Over Satellite





Requirements

- No Wi-Fi or cellular coverage
- SIM installed
- Friends added while online

Demo Mode

Requirements

- Satellite service available on the specific iPhone

Internally called "Emergency Try Out"

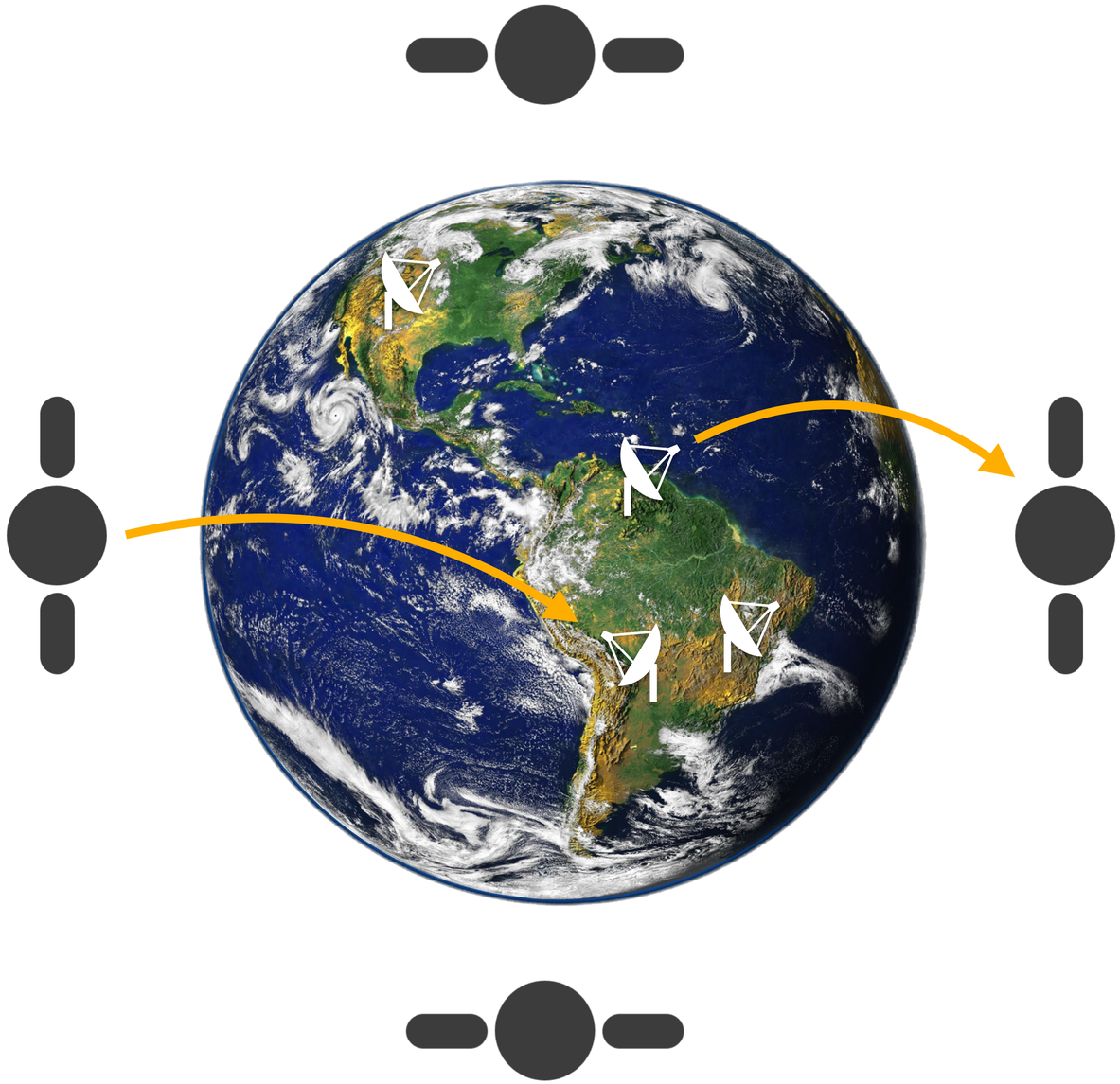
Availability

iOS 16.1

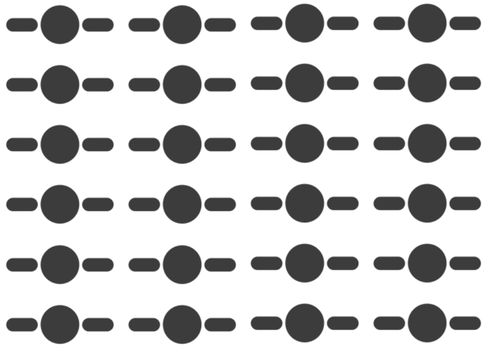
iOS 16.2

iOS 16.4

Globalstar Satellite Network



Stats



Minimum 28 Satellites

1400 km (870 miles) altitude

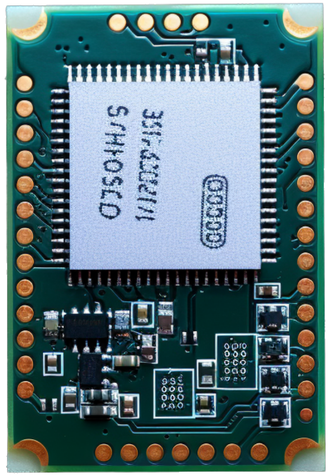
24 ground stations

Closest one in France

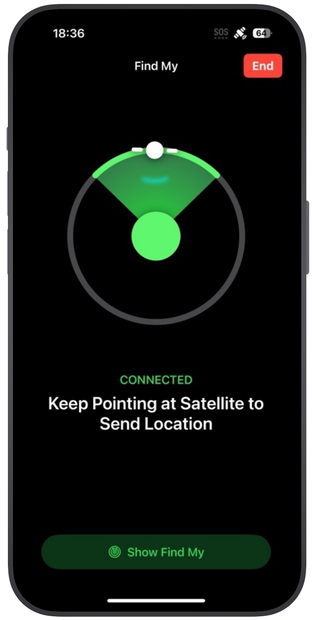
Provided Services

Phones

Messengers

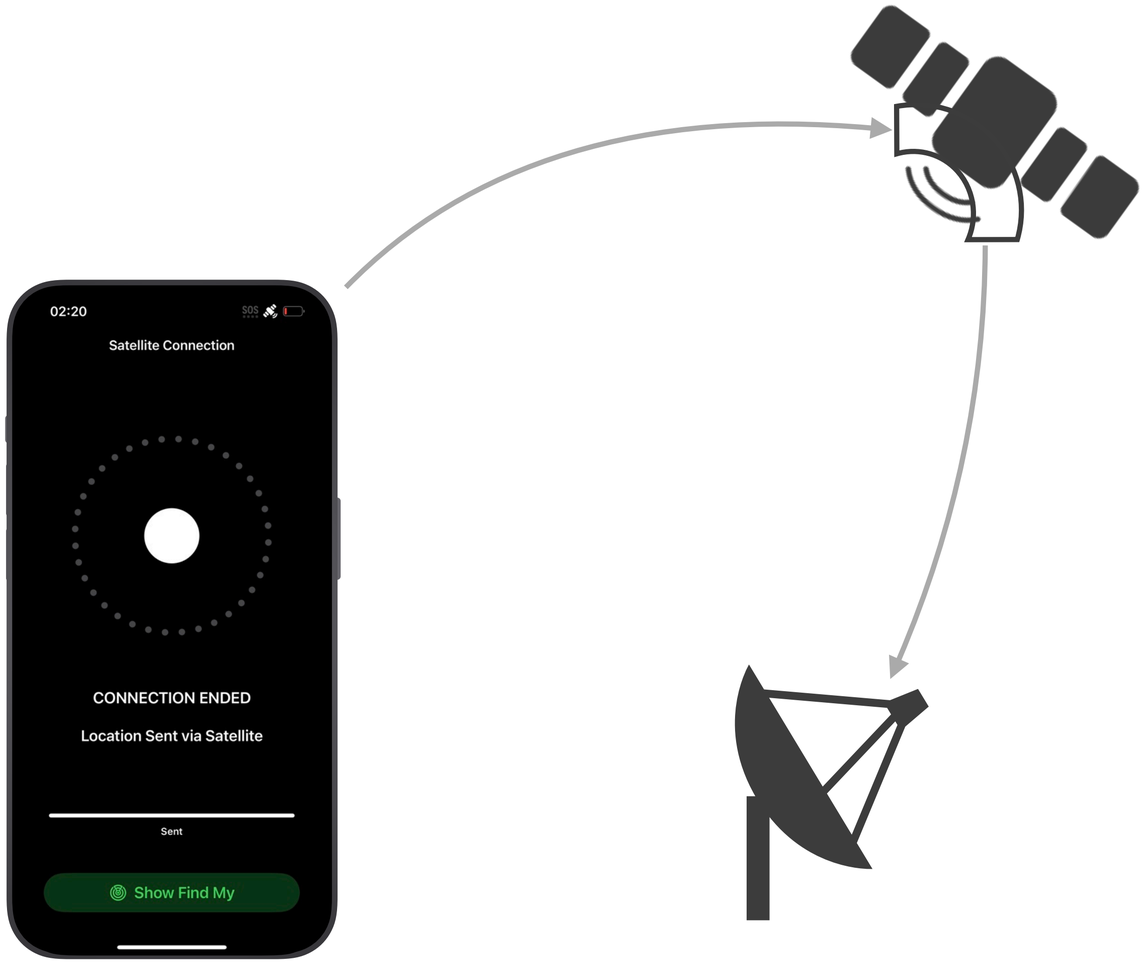


IoT Chips



iPhones

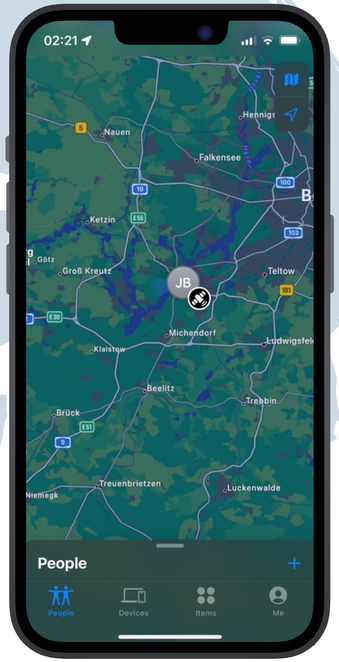
Satellite Communication Basis

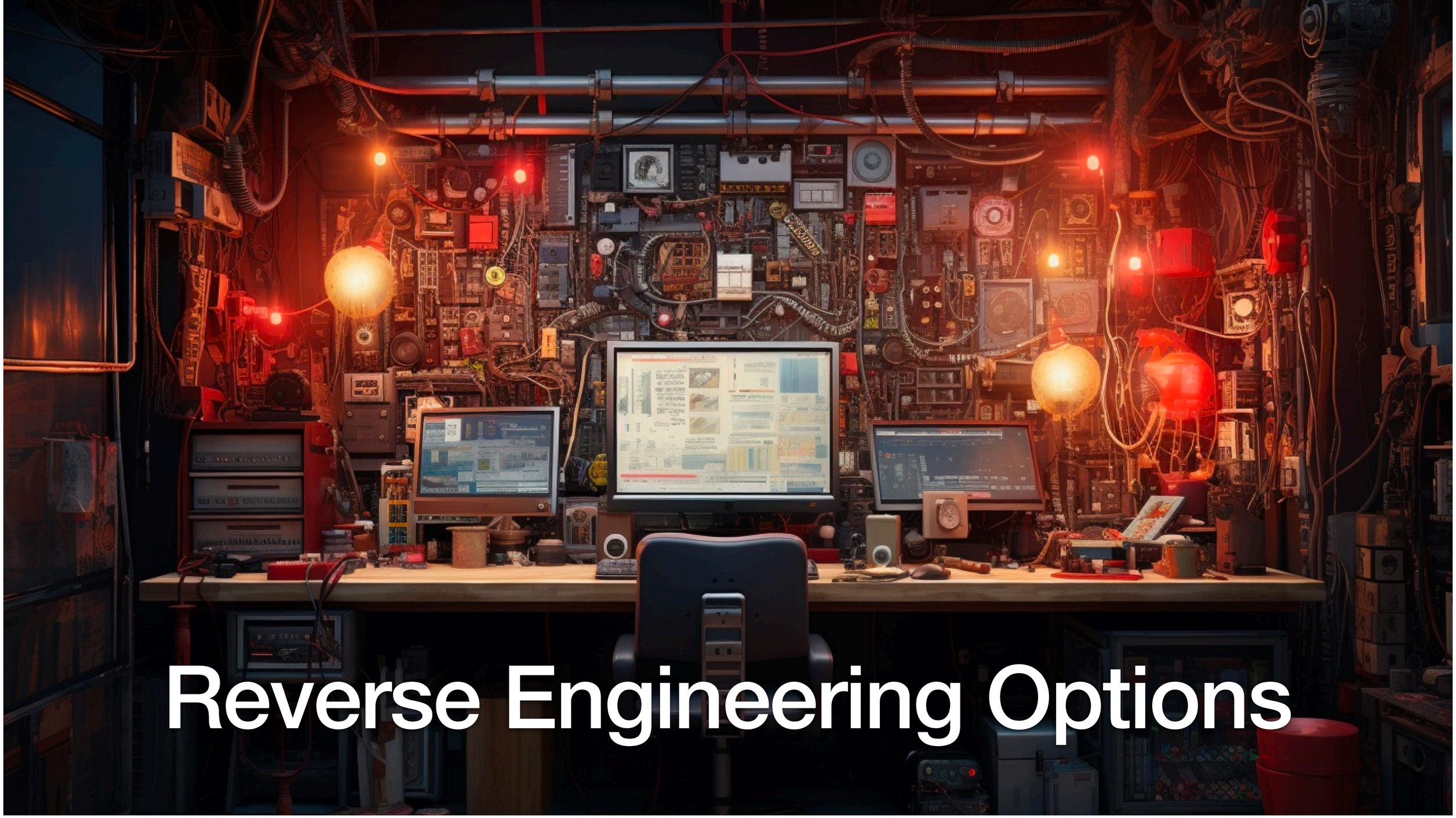


- Bent-pipe satellites
- Messages are reflected
- Decoding at the ground station
- Maximum 9600bps

Target

Anchor

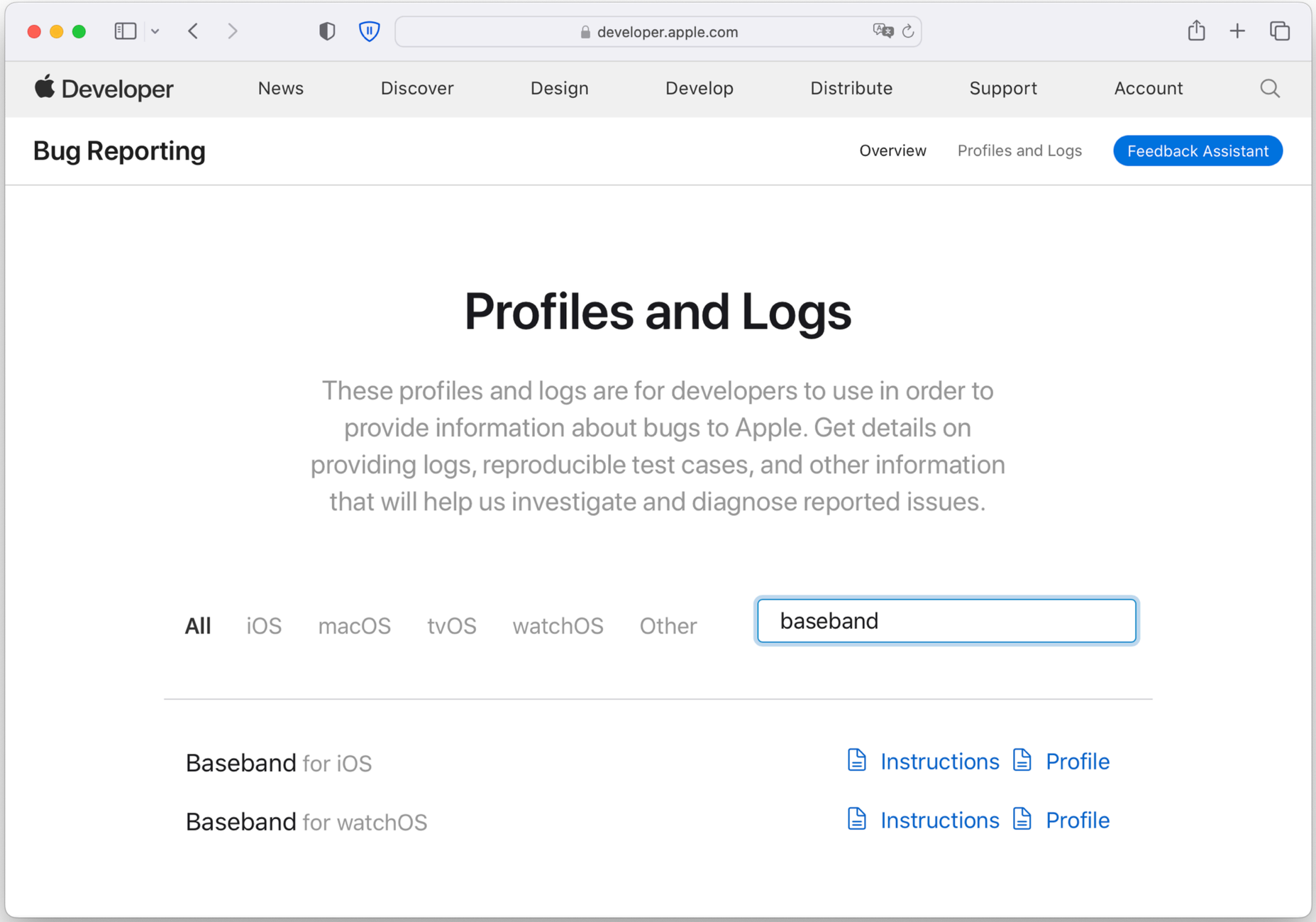


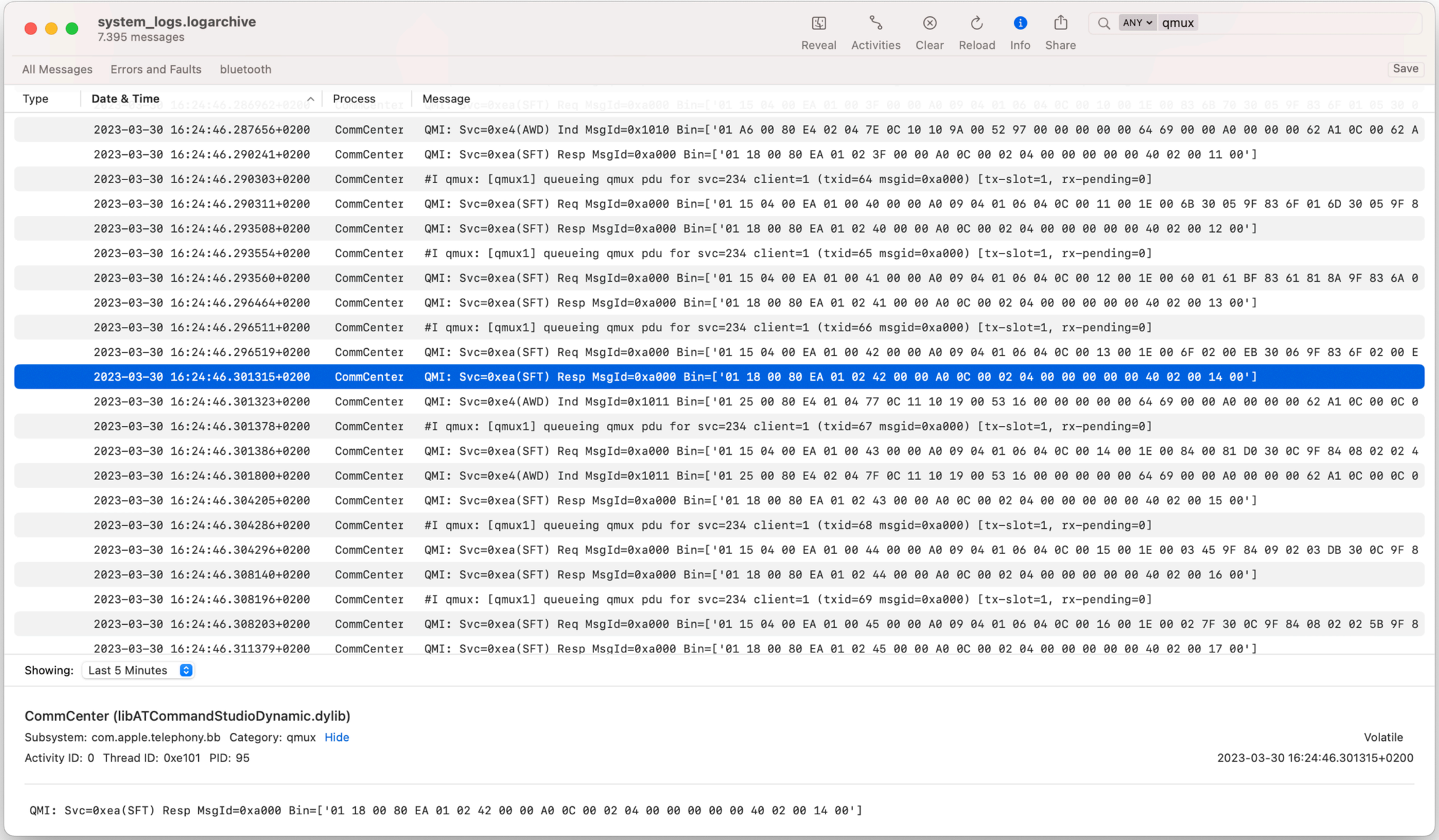


Reverse Engineering Options

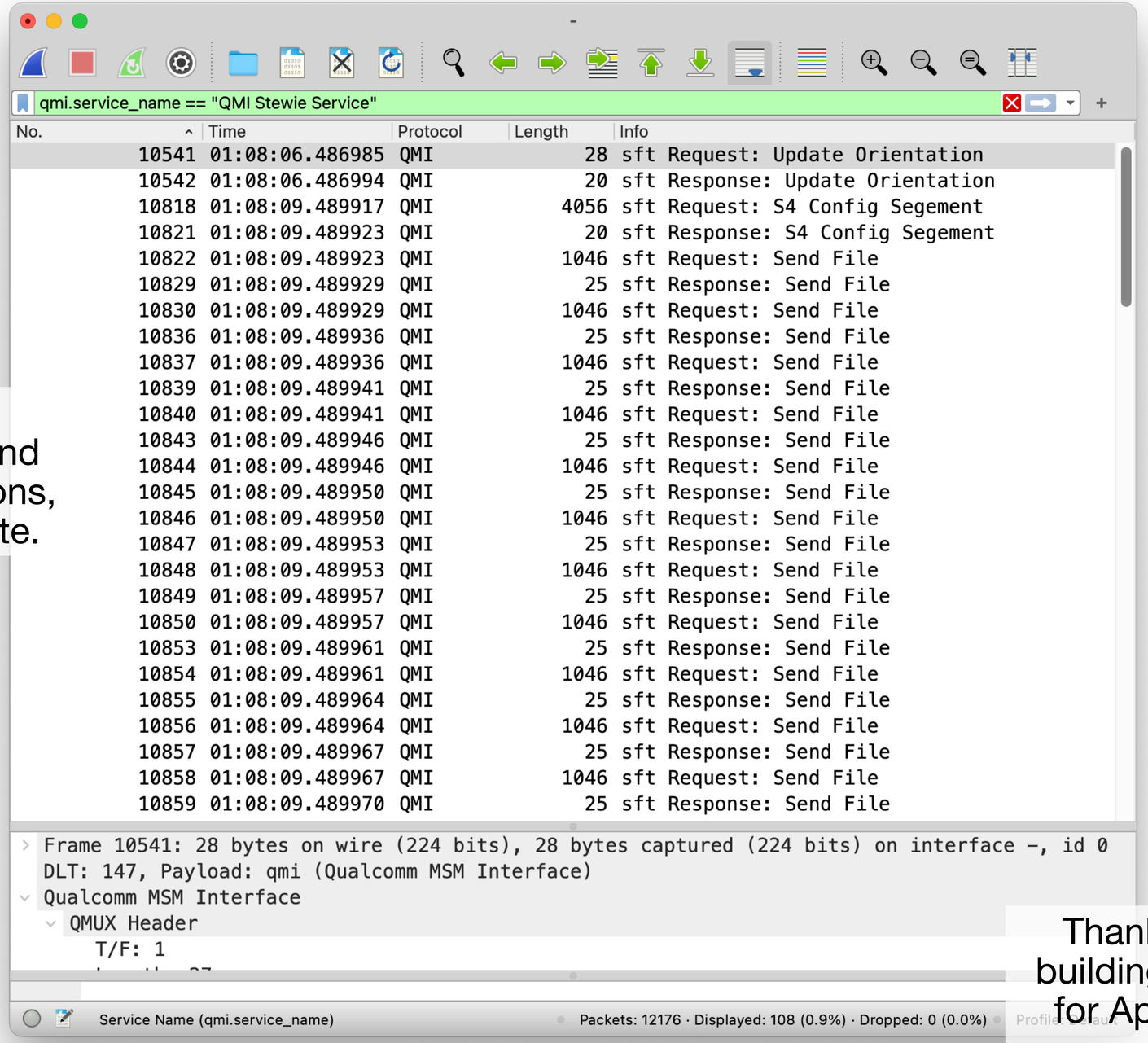
ios 16 jelbrek for iphone 14, wen eta?

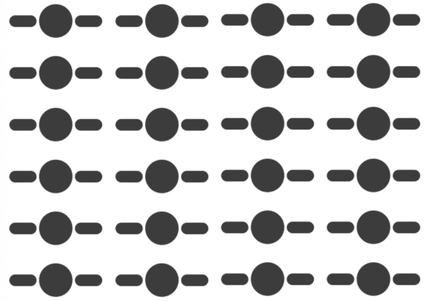

Static reversing & staring on iPhone 14 logs...


developer.apple.com
Developer
News
Discover
Design
Develop
Distribute
Support
Account
Bug Reporting
Overview
Profiles and Logs
Feedback Assistant
Profiles and Logs
These profiles and logs are for developers to use in order to provide information about bugs to Apple. Get details on providing logs, reproducible test cases, and other information that will help us investigate and diagnose reported issues.
All
iOS
macOS
tvOS
watchOS
Other
baseband
Baseband for iOS
Instructions
Profile
Baseband for watchOS
Instructions
Profile


system_logs.logarchive
7.395 messages
Reveal
Activities
Clear
Reload
Info
Share
ANY
qmux
All Messages
Errors and Faults
bluetooth
Save
Type
Date & Time
Process
Message
2023-03-30 16:24:46.287656+0200 CommCenter QMI: Svc=0xe4(AWD) Ind MsgId=0x1010 Bin=['01 A6 00 80 E4 02 04 7E 0C 10 10 9A 00 52 97 00 00 00 00 00 64 69 00 00 A0 00 00 00 62 A1 0C 00 62 A
2023-03-30 16:24:46.290241+0200 CommCenter QMI: Svc=0xea(SFT) Resp MsgId=0xa000 Bin=['01 18 00 80 EA 01 02 3F 00 00 A0 0C 00 02 04 00 00 00 00 00 40 02 00 11 00']
2023-03-30 16:24:46.290303+0200 CommCenter #I qmux: [qmux1] queueing qmux pdu for svc=234 client=1 (txid=64 msgid=0xa000) [tx-slot=1, rx-pending=0]
2023-03-30 16:24:46.290311+0200 CommCenter QMI: Svc=0xea(SFT) Req MsgId=0xa000 Bin=['01 15 04 00 EA 01 00 40 00 00 A0 09 04 01 06 04 0C 00 11 00 1E 00 6B 30 05 9F 83 6F 01 6D 30 05 9F 8
2023-03-30 16:24:46.293508+0200 CommCenter QMI: Svc=0xea(SFT) Resp MsgId=0xa000 Bin=['01 18 00 80 EA 01 02 40 00 00 A0 0C 00 02 04 00 00 00 00 00 40 02 00 12 00']
2023-03-30 16:24:46.293554+0200 CommCenter #I qmux: [qmux1] queueing qmux pdu for svc=234 client=1 (txid=65 msgid=0xa000) [tx-slot=1, rx-pending=0]
2023-03-30 16:24:46.293560+0200 CommCenter QMI: Svc=0xea(SFT) Req MsgId=0xa000 Bin=['01 15 04 00 EA 01 00 41 00 00 A0 09 04 01 06 04 0C 00 12 00 1E 00 60 01 61 BF 83 61 81 8A 9F 83 6A 0
2023-03-30 16:24:46.296464+0200 CommCenter QMI: Svc=0xea(SFT) Resp MsgId=0xa000 Bin=['01 18 00 80 EA 01 02 41 00 00 A0 0C 00 02 04 00 00 00 00 00 40 02 00 13 00']
2023-03-30 16:24:46.296511+0200 CommCenter #I qmux: [qmux1] queueing qmux pdu for svc=234 client=1 (txid=66 msgid=0xa000) [tx-slot=1, rx-pending=0]
2023-03-30 16:24:46.296519+0200 CommCenter QMI: Svc=0xea(SFT) Req MsgId=0xa000 Bin=['01 15 04 00 EA 01 00 42 00 00 A0 09 04 01 06 04 0C 00 13 00 1E 00 6F 02 00 EB 30 06 9F 83 6F 02 00 E
2023-03-30 16:24:46.301315+0200 CommCenter QMI: Svc=0xea(SFT) Resp MsgId=0xa000 Bin=['01 18 00 80 EA 01 02 42 00 00 A0 0C 00 02 04 00 00 00 00 00 40 02 00 14 00']
2023-03-30 16:24:46.301323+0200 CommCenter QMI: Svc=0xe4(AWD) Ind MsgId=0x1011 Bin=['01 25 00 80 E4 01 04 77 0C 11 10 19 00 53 16 00 00 00 00 00 64 69 00 00 A0 00 00 00 62 A1 0C 00 0C 0
2023-03-30 16:24:46.301378+0200 CommCenter #I qmux: [qmux1] queueing qmux pdu for svc=234 client=1 (txid=67 msgid=0xa000) [tx-slot=1, rx-pending=0]
2023-03-30 16:24:46.301386+0200 CommCenter QMI: Svc=0xea(SFT) Req MsgId=0xa000 Bin=['01 15 04 00 EA 01 00 43 00 00 A0 09 04 01 06 04 0C 00 14 00 1E 00 84 00 81 D0 30 0C 9F 84 08 02 02 4
2023-03-30 16:24:46.301800+0200 CommCenter QMI: Svc=0xe4(AWD) Ind MsgId=0x1011 Bin=['01 25 00 80 E4 02 04 7F 0C 11 10 19 00 53 16 00 00 00 00 00 64 69 00 00 A0 00 00 00 62 A1 0C 00 0C 0
2023-03-30 16:24:46.304205+0200 CommCenter QMI: Svc=0xea(SFT) Resp MsgId=0xa000 Bin=['01 18 00 80 EA 01 02 43 00 00 A0 0C 00 02 04 00 00 00 00 00 40 02 00 15 00']
2023-03-30 16:24:46.304286+0200 CommCenter #I qmux: [qmux1] queueing qmux pdu for svc=234 client=1 (txid=68 msgid=0xa000) [tx-slot=1, rx-pending=0]
2023-03-30 16:24:46.304296+0200 CommCenter QMI: Svc=0xea(SFT) Req MsgId=0xa000 Bin=['01 15 04 00 EA 01 00 44 00 00 A0 09 04 01 06 04 0C 00 15 00 1E 00 03 45 9F 84 09 02 03 DB 30 0C 9F 8
2023-03-30 16:24:46.308140+0200 CommCenter QMI: Svc=0xea(SFT) Resp MsgId=0xa000 Bin=['01 18 00 80 EA 01 02 44 00 00 A0 0C 00 02 04 00 00 00 00 00 40 02 00 16 00']
2023-03-30 16:24:46.308196+0200 CommCenter #I qmux: [qmux1] queueing qmux pdu for svc=234 client=1 (txid=69 msgid=0xa000) [tx-slot=1, rx-pending=0]
2023-03-30 16:24:46.308203+0200 CommCenter QMI: Svc=0xea(SFT) Req MsgId=0xa000 Bin=['01 15 04 00 EA 01 00 45 00 00 A0 09 04 01 06 04 0C 00 16 00 1E 00 02 7F 30 0C 9F 84 08 02 02 5B 9F 8
2023-03-30 16:24:46.311379+0200 CommCenter QMI: Svc=0xea(SFT) Resp MsgId=0xa000 Bin=['01 18 00 80 EA 01 02 45 00 00 A0 0C 00 02 04 00 00 00 00 00 40 02 00 17 00']
Showing:
Last 5 Minutes
CommCenter (libATCommandStudioDynamic.dylib)
Subsystem: com.apple.telephony.bb Category: qmux Hide
Activity ID: 0 Thread ID: 0xe101 PID: 95
Volatile
2023-03-30 16:24:46.301315+0200
QMI: Svc=0xea(SFT) Resp MsgId=0xa000 Bin=['01 18 00 80 EA 01 02 42 00 00 A0 0C 00 02 04 00 00 00 00 00 40 02 00 14 00']

Send current device orientation to modem, and then send satellite positions, one file per active satellite.





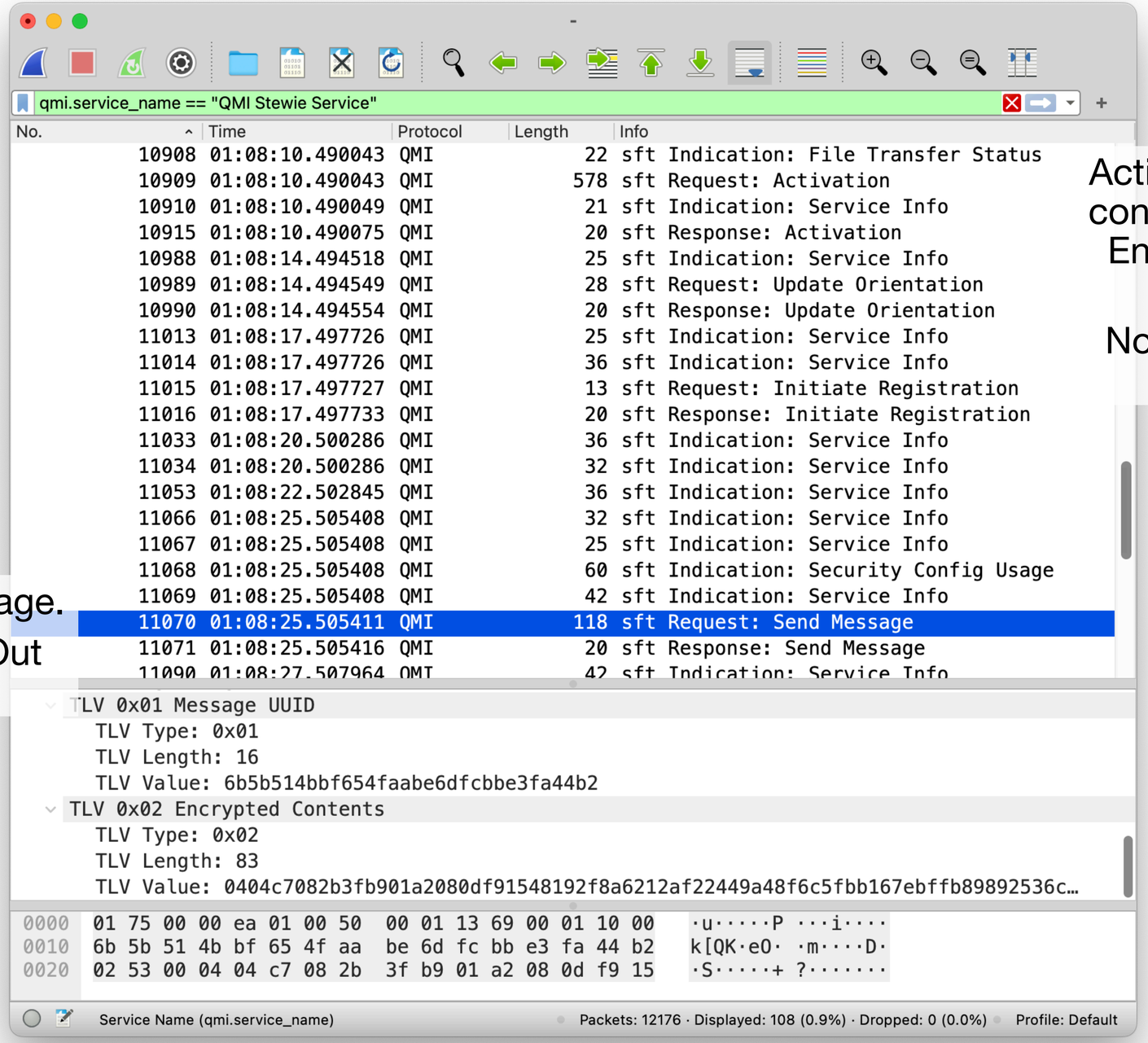
Thanks @Lukas Arnold for building Wireshark dissectors for Apple's proprietary QMI services!

Activation + Security Config, containing Ephemeral Public Encryption Key (EPKI) and further data.

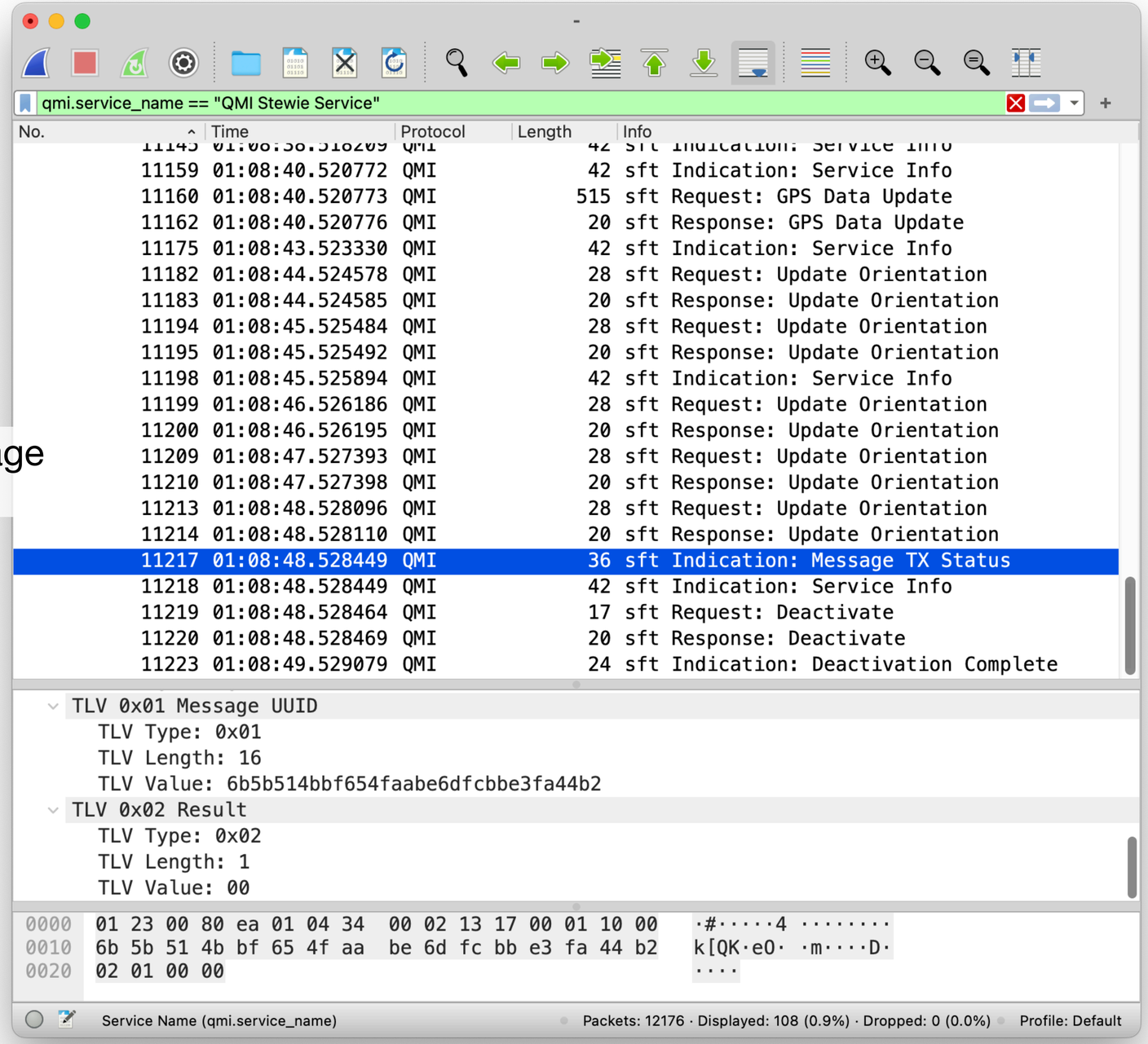
Not present during Try Out mode!

Send an encrypted message.

Not present during Try Out mode!

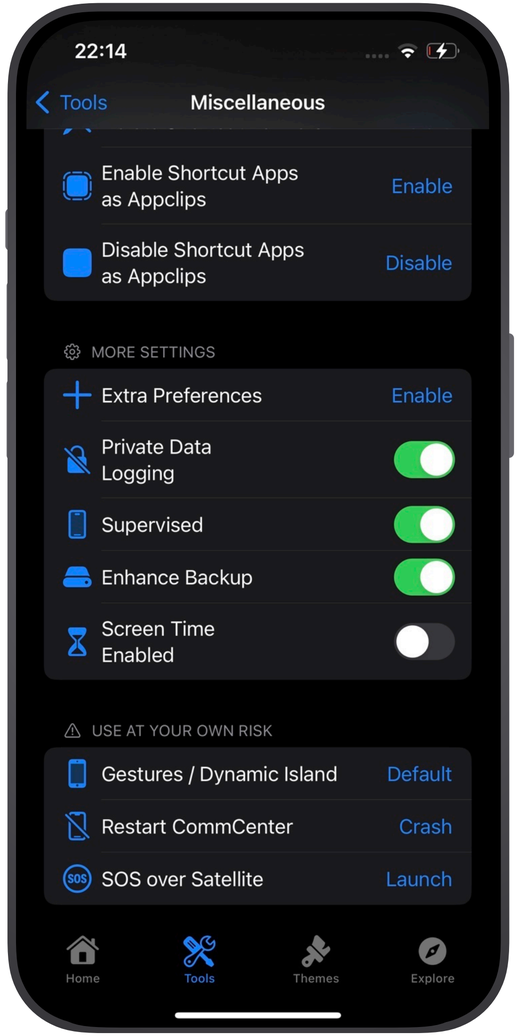


Indicator that the message was sent.



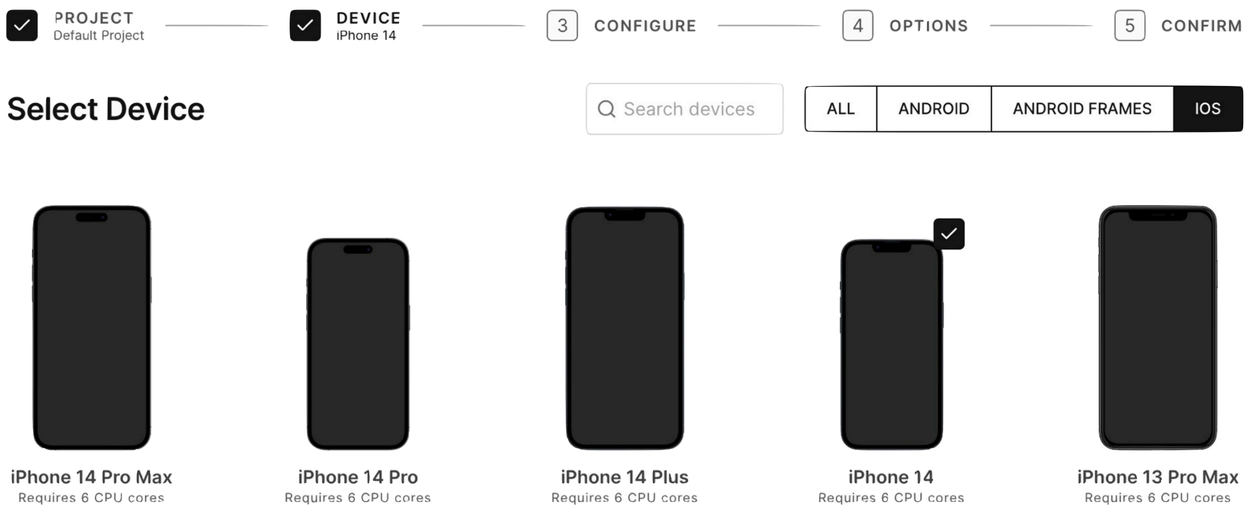
Cowabunga

- Works on iPhone 14 with iOS 16.1.2
- Made some custom tweaks
  - Enhanced logging even when no profile is available
  - Backup more data
  - Crash CommCenter to observe initialisation



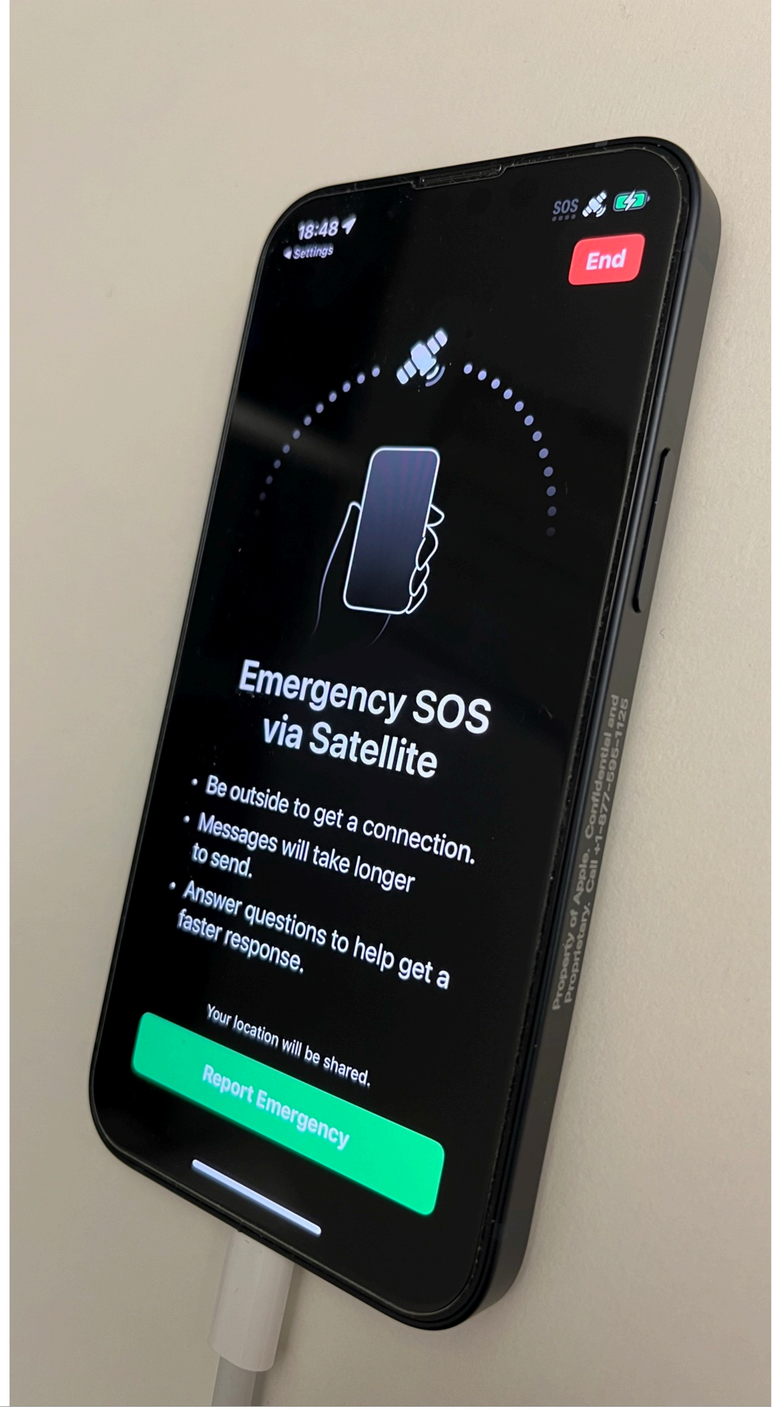
Corellium

- Jailbroken iOS 16 on iPhone 14
- Partially thinks it has satellite features
- Easy dynamic instrumentation

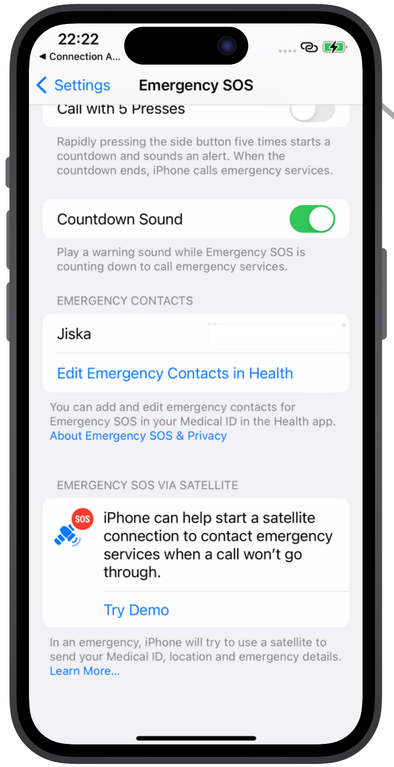


Security Research Device

- iPhone 13 mini
- Modem doesn't actually support satellite services
- Most binaries required for satellite services still present
- Can we make it talk to a satellite?




Internal Components



XPC

`com.apple.commcenter.coretelephony.xpc`

`requestStewieWithContext, reason=5 (EmergencyTryOut)`

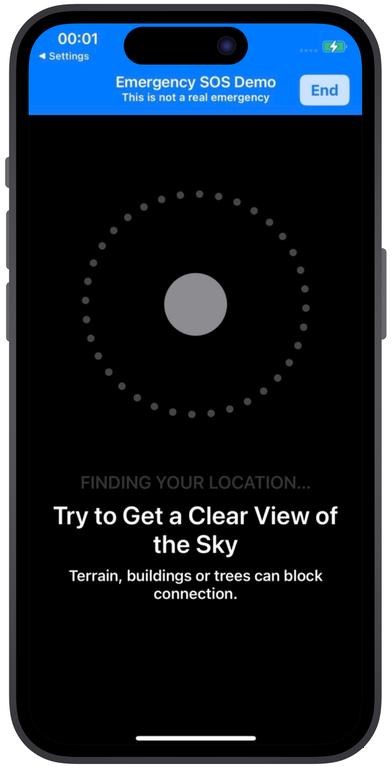
Launch via URL

`x-apple-sosbuddy://request?reason=OfferEmergencyTryOut`

Preferences

[?] CommCenter [?]

SOSBuddy



XPC

com.apple.commcenter.coretelephony.xpc

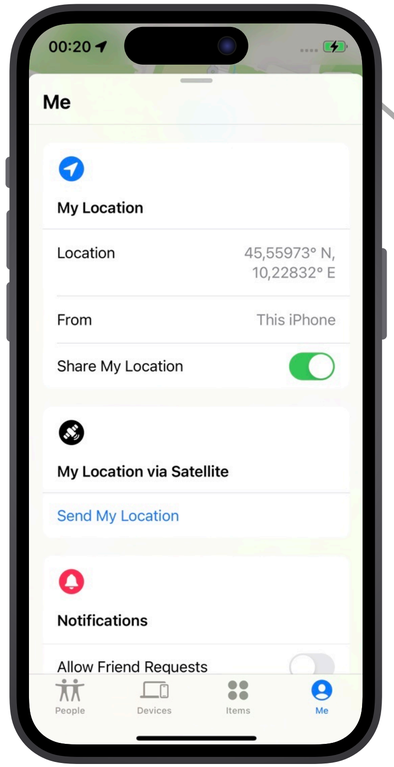
reportStewieDeviceOrientation

CommCenter

SOSBuddy

com.apple.commcenter.coretelephony.xpc

connectionAssistantStateChanged



searchpartyd

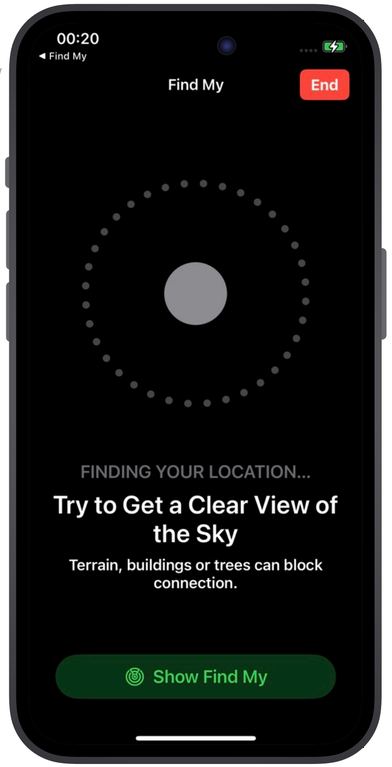
XPC

`com.apple.commcenter.coretelephony.xpc`

`requestStewieWithContext, reason=6 (FindMy)`

Launch via URL

`x-apple-sosbuddy://request?reason=FindMy`



Find My

[?] CommCenter [?]

SOSBuddy

Pizza icon!

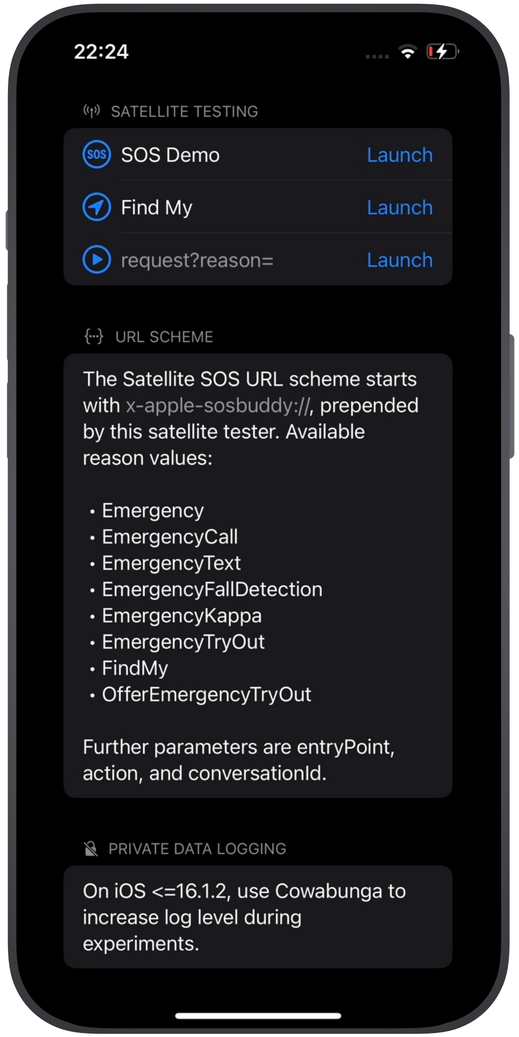
CommCenter will update involved parties upon connection updates.

Calling URLs directly?

Reverse engineer CommCenter, build an app for non-jailbroken iPhone 14, ...

`reason=`

0. `Unknown`
1. `EmergencyCall`
2. `EmergencyText`
3. `EmergencyFallDetection`
4. `EmergencyKappa`
5. `EmergencyTryOut`
6. `OfferEmergencyTryOut`
7. `FindMy`
8. `Test`
9. `Anywhere`
10. `AnywhereTest`



Only works for EmergencyTryOut.

Other services are not active and are missing state in CommCenter.

Setting custom Stewie context with reason!

Launch via URL

```
x-apple-sosbuddy://request?
reason=EmergencyText&conversationId=0
&action=startConversation
```

XPC

```
com.apple.commcenter.coretelephony.xpc
  requestStewieWithContext, reason=2
            (EmergencyText)
```

Frida

[?] CommCenter [?]

SOSBuddy

Fake it 'till you make it!

iPhone 14

No jailbreak but modem with satellite

iPhone 13 mini

Research device but no satellite modem

/private/var/containers/Shared/SystemGroup/systemgroup.com.apple.mobilegestaltcache/Library/Caches/com.apple.MobileGestalt.plist

libMobileGestalt.dylib

Mobile Device Properties

```
hwconfig=0x34
modelAP="D27AP"
modelPhone="iPhone 14,7"
```

CommCenter

libCommCenterBase.dylib

Initialisation

`GsmRadioPersonality::create`

```
GsmRadioPersonality*
  GsmRadioPersonality::GsmRadioPersonality(
    GsmRadioPersonality *this,
    shared_ptr *ptr,
    HardwareConfiguration hwconfig,
    basic_string *modelAP,
    basic_string *modelPhone)

  //...

  if (hwconfig >= 0x34) {
    this->bifrost_enabled_by_config =
          os_feature_enabled_impl("CoreTelephony","Bifrost");
    this->bifrost_compatible = 1;

    // ...

  }
}
```

System Features

`Bifrost`

libsystem_featureflags.dylib

/System/Library/FeatureFlags/Domain/CoreTelephony.plist

Values as observed on Corellium iPhone 14.

CommCenter

~~Parse CLI Options~~

```
sleep(0x10)
sleep(0x10)
```

Attach Frida

Binary patching!

/private/var/containers/Shared/SystemGroup/systemgroup.com.apple.mobilegestaltcache/Library/Caches/com.apple.MobileGestalt.plist

libMobileGestalt.dylib

Overwriting this will screw up CommCenter state, it will try to load iPhone 14 baseband firmware etc.

libCommCenterBase.dylib

```
const GsmRadioPersonality_create = Module.getExportByName(null,
'_ZN19GsmRadioPersonality6createERKNSt3__110shared_ptrIK8RegistryEE21Har
dwareConfigurationRKNS0_12basic_stringIcNS0_11char_traitsIcEENS0_9alloca
torIcEEEESF_');

Interceptor.attach(GsmRadioPersonality_create, {
  onEnter(args) {
    console.log(`GsmRadioPersonality::create called with hw_model=
                ${args[1]}, replacing with 0x34 for iPhone 14!`);
    args[1] = new NativePointer(0x34);
  }
});
```

Mobile Device Properties

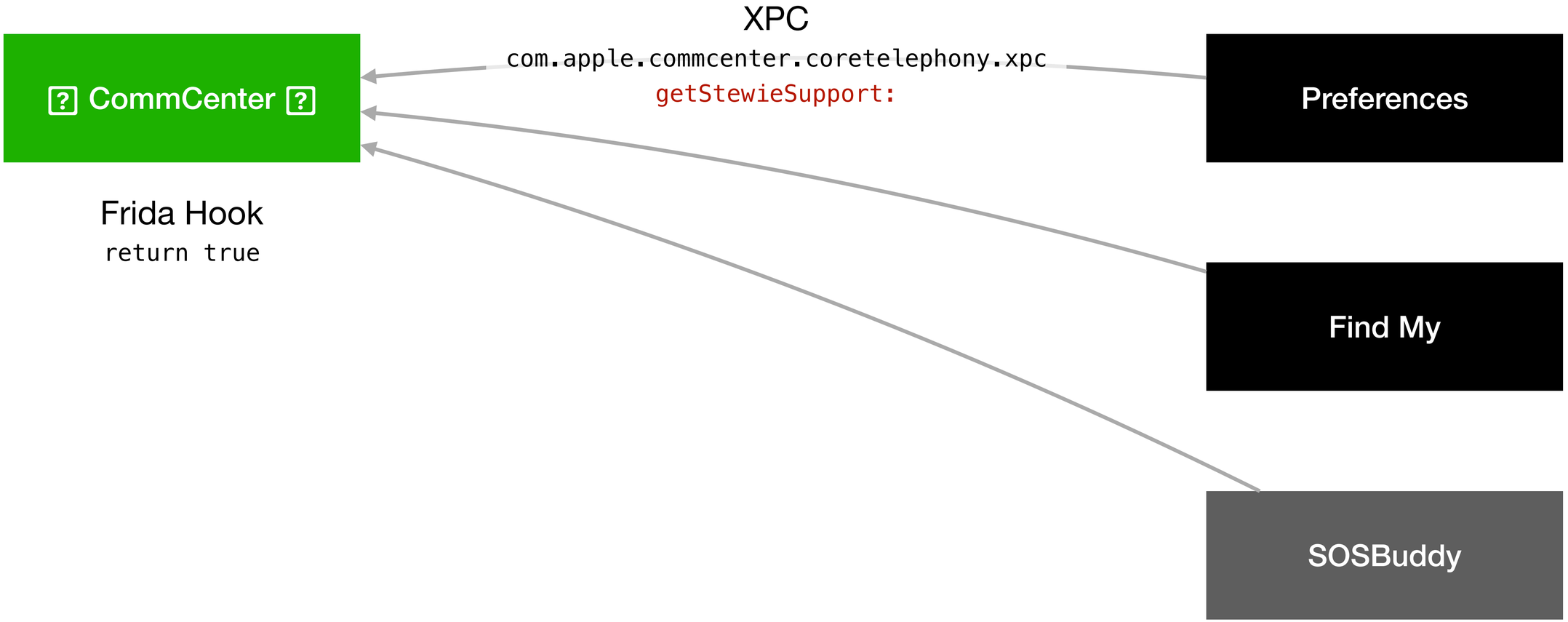
```
hwconfig=0x34
```

libsystem_featureflags.dylib

Already present on SRD.

/System/Library/FeatureFlags/Domain/CoreTelephony.plist

Applied to SRD!


XPC
com.apple.commcenter.coretelephony.xpc
getStewieSupport:
CommCenter
Frida Hook
return true
Preferences
Find My
SOSBuddy

No Cellular, no Wi-Fi?

`[CTStewieState setAllowedServices:0xc006]`

...a few Frida hooks later, CommCenter will always allow all services, even outside a launch country.

Service Mask

Mask Bit	Meaning
0x8000	Anywhere
0x4000	Test
0x0004	FindMy
0x0002	EmergencyTryOut
0x0001	Default

```
Interceptor.attach(requestStewieForService, {
    onEnter(args) {
        console.log(`Allowing requestStewieForService`);

        // allow all services in the 2-bytes service mask
        args[0].add(0xf2).writeShort(0xffff);

        // also allow the service right now
        args[0].add(0xf0).writeU8(0x1);
    }
});
```

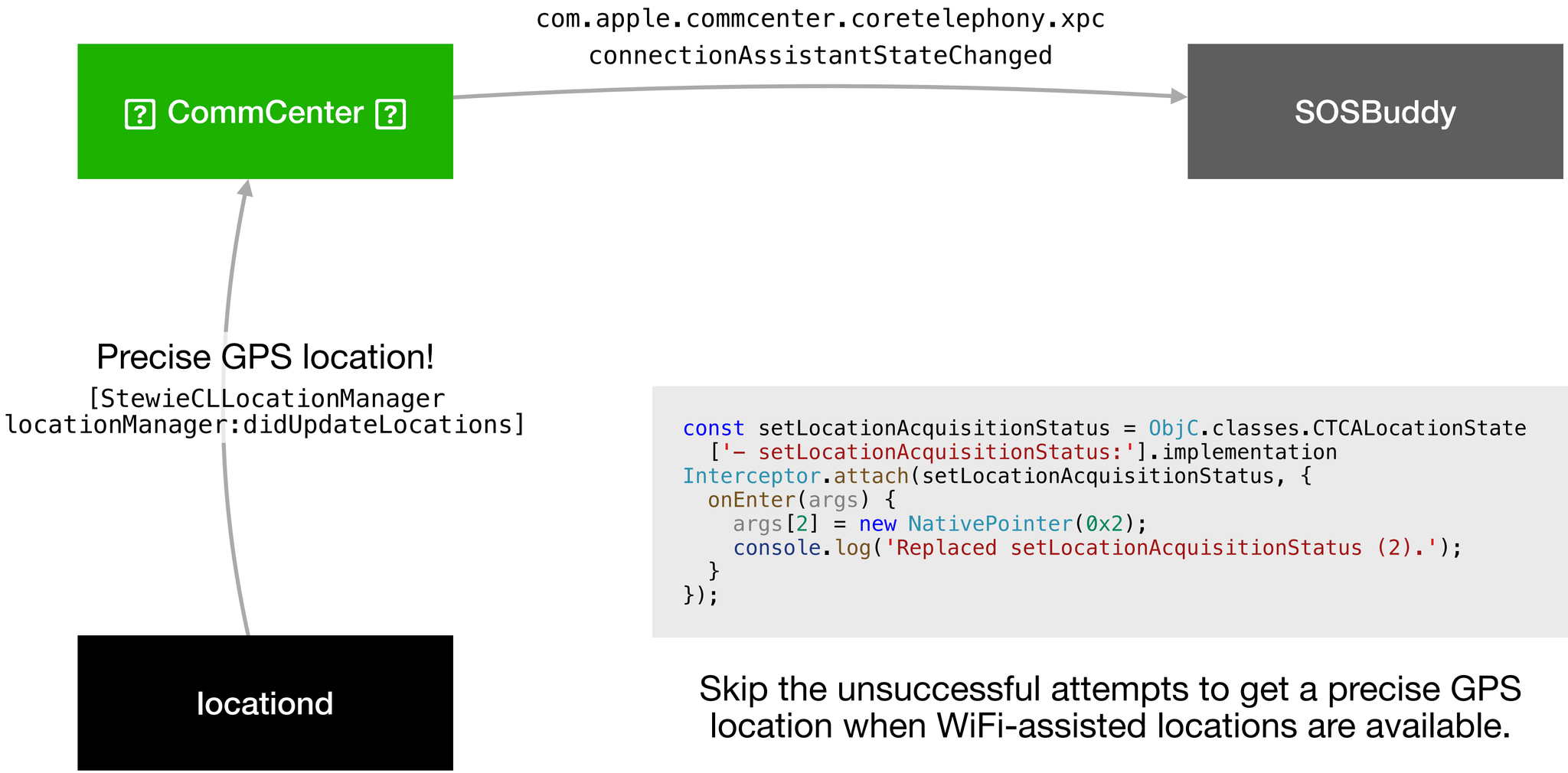
Manually call SOSBuddy!

```
function openURL(url) {
   var w = ObjC.classes.LSApplicationWorkspace.defaultWorkspace();
   var toOpen = ObjC.classes.NSURL.URLWithString_(url);
   return w.openSensitiveURL_withOptions_(toOpen, null);
}

openURL("x-apple-sosbuddy://request?reason=OfferEmergencyTryOut")
```

Requires iOS built with SOSBuddy.

Launch from SpringBoard with Frida.



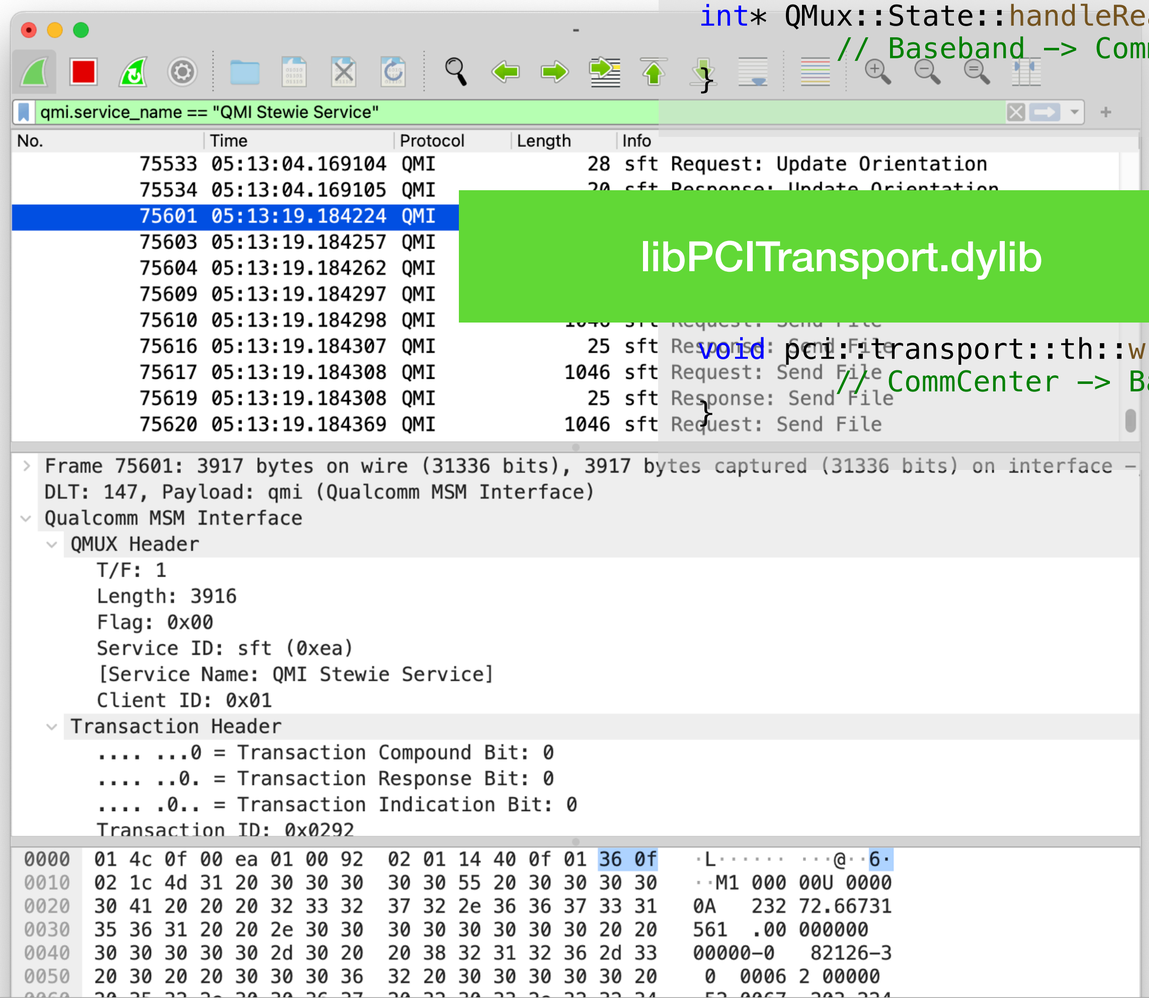

```
const setLocationAcquisitionStatus = ObjC.classes.CTCALocationState
  ['- setLocationAcquisitionStatus:'].implementation
Interceptor.attach(setLocationAcquisitionStatus, {
  onEnter(args) {
    args[2] = new NativePointer(0x2);
    console.log('Replaced setLocationAcquisitionStatus (2).');
  }
});
```

Skip the unsuccessful attempts to get a precise GPS location when WiFi-assisted locations are available.

? CommCenter ?

libATCommandStudioDynamic.dylib

```
int* QMux::State::handleReadData(State *this, char *buffer, uint len) {
    // Baseband -> CommCenter packets
}
```



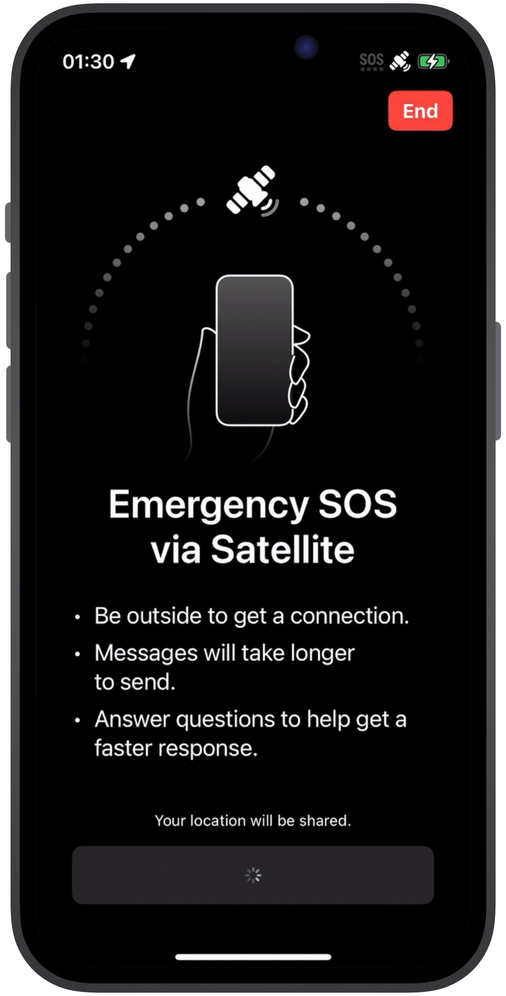
libPCITransport.dylib

```
void pci::transport::th::writeAsync(State *this, char *buffer, uint len, void *callback) {
    // CommCenter -> Baseband packets
}
```

Overwrite messages from the baseband when it rejects packets.

Inject new packets where needed.

We fake everything including satellite configs and authentication.



Enabling Stewie on SRD allows us to analyse the satellite ecosystem in action!

Stewie & Bifröst Configuration

What happens on-device when Stewie is available?

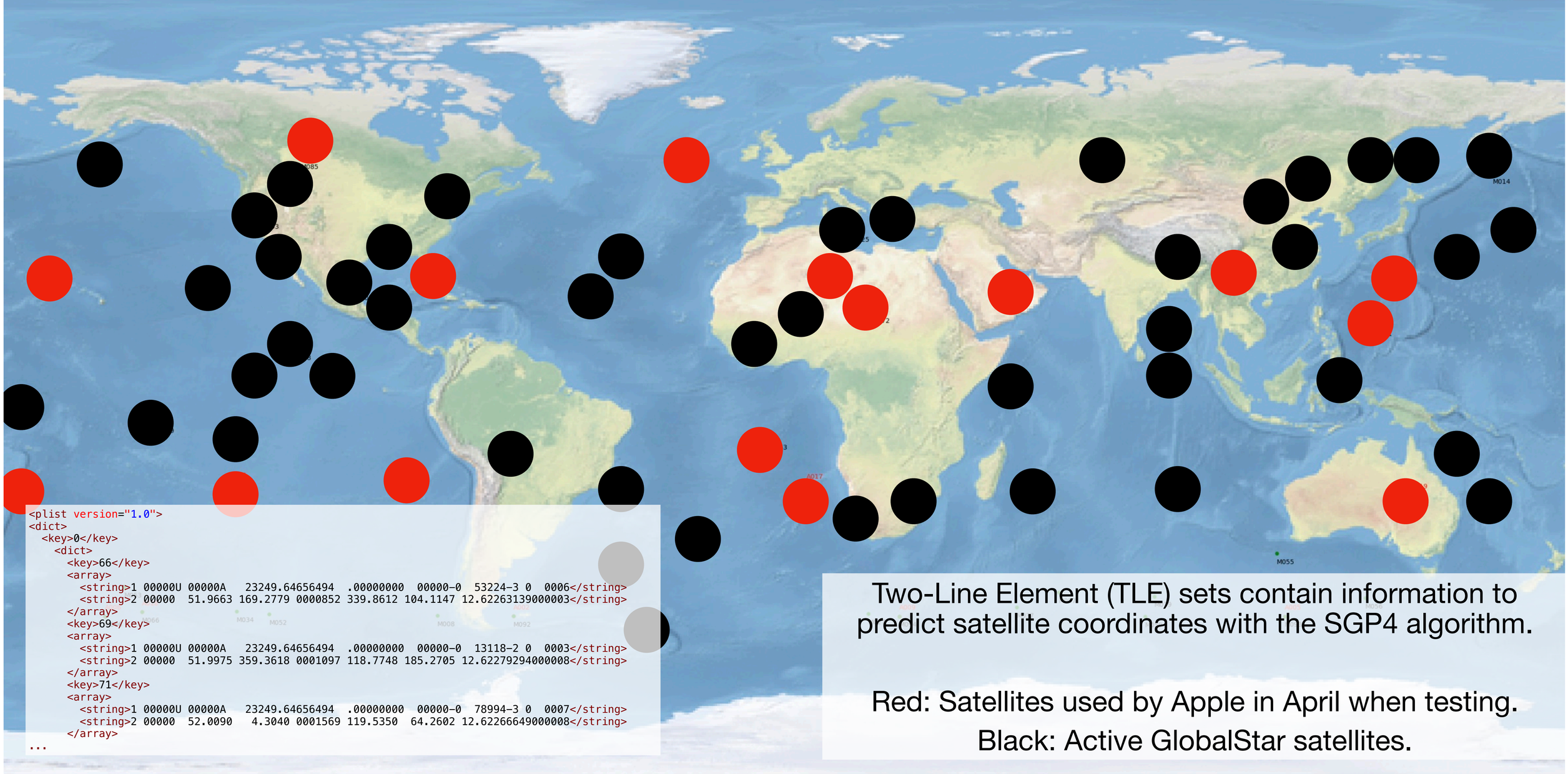
Entitlements

```
<key>com.apple.trial.client</key>
<array>
    <string>800</string>
    <string>801</string>
    <string>802</string>
    <string>803</string>
    ...
</array>
```

`/private/var/mobile/Library/Trial/Treatments/802/factorPacks/.../assets/Targets/Targets.plist`

`/private/var/mobile/Library/Trial/Treatments/803/factorPacks/.../assets/Config/Config.plist`

CommCenter only downloads these if Stewie is available.




```
<plist version="1.0">
<dict>
  <key>0</key>
    <dict>
      <key>66</key>
      <array>
        <string>1 00000U 00000A   23249.64656494  .00000000  00000-0  53224-3 0  0006</string>
        <string>2 00000  51.9663 169.2779 0000852 339.8612 104.1147 12.62263139000003</string>
      </array>
      <key>69</key>
      <array>
        <string>1 00000U 00000A   23249.64656494  .00000000  00000-0  13118-2 0  0003</string>
        <string>2 00000  51.9975 359.3618 0001097 118.7748 185.2705 12.62279294000008</string>
      </array>
      <key>71</key>
      <array>
        <string>1 00000U 00000A   23249.64656494  .00000000  00000-0  78994-3 0  0007</string>
        <string>2 00000  52.0090   4.3040 0001569 119.5350  64.2602 12.62266649000008</string>
      </array>
...
```

Two-Line Element (TLE) sets contain information to predict satellite coordinates with the SGP4 algorithm.

Red: Satellites used by Apple in April when testing.

Black: Active GlobalStar satellites.

`/private/var/mobile/Library/Trial/Treatments/802/factorPacks/.../assets/Targets/Targets.plist`

Thanks @Fabian Portner for figuring this out :)

```
<plist version="1.0">
<dict>
  <key>0</key>
  <dict>
    <key>countries</key>
    <array>
      <dict>
        <key>fwd_alternate_channels</key>
        <array>
          <integer>262220</integer>
          <integer>262270</integer>
        </array>
        <key>fwd_channel</key>
        <integer>262170</integer>
        <key>iso3166_alpha_3</key>
        <string>CAN</string>
        <key>rev_channels</key>
        <dict>
          <key>nominal</key>
          <array>
            <integer>262336</integer>
            <integer>262338</integer>
            <integer>262340</integer>
            <integer>262342</integer>
            <integer>262344</integer>
            <integer>262346</integer>
            <integer>262348</integer>
```

Channel numbers, not frequencies.

/private/var/mobile/Library/Trial/Treatments/803/factorPacks/.../assets/Config/Config.plist

XPC

com.apple.mobileasset.autoasset

assetType=com.apple.MobileAsset.BifrostCodecs,
assetSpecifier=English

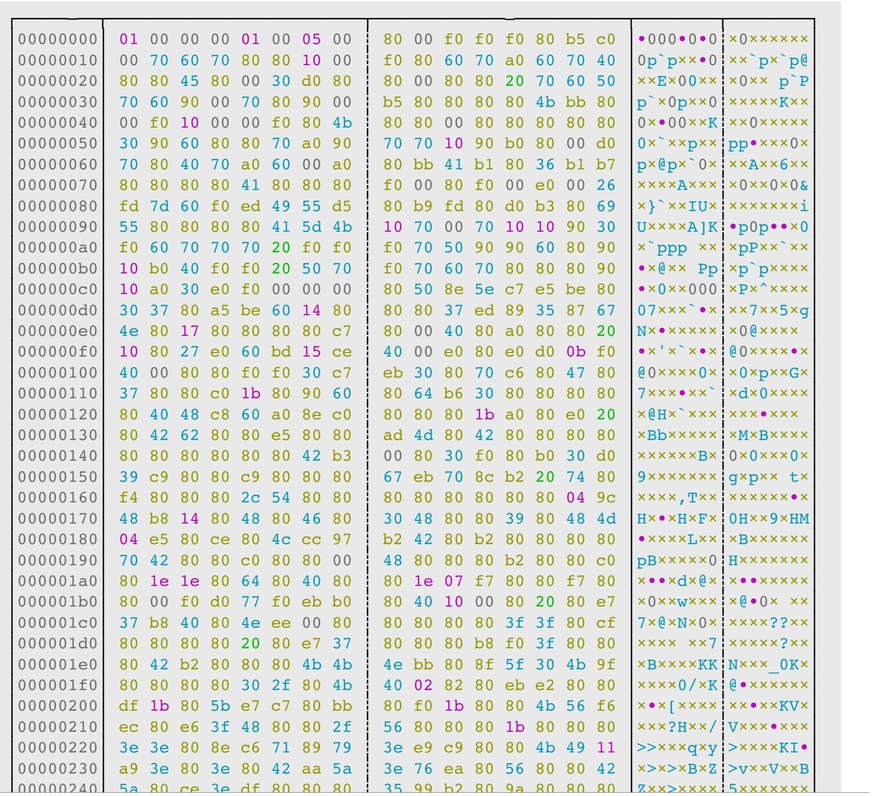
CommCenter

mobileassetd

Entitlements

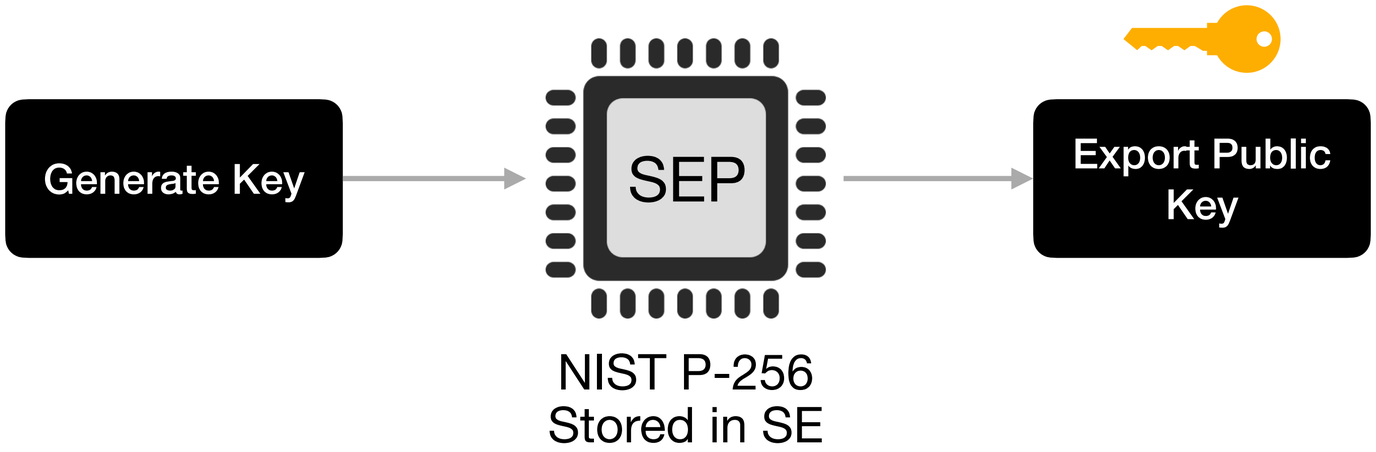
```
<key>com.apple.private.assets.accessible-asset-types</key>
<array>
  <string>com.apple.MobileAsset.BifrostCodecs</string>
</array>
```

/private/var/MobileAsset/AssetsV2/com_apple_MobileAsset_BifrostCodecs/purpose_auto/...asset/AssetData/codec

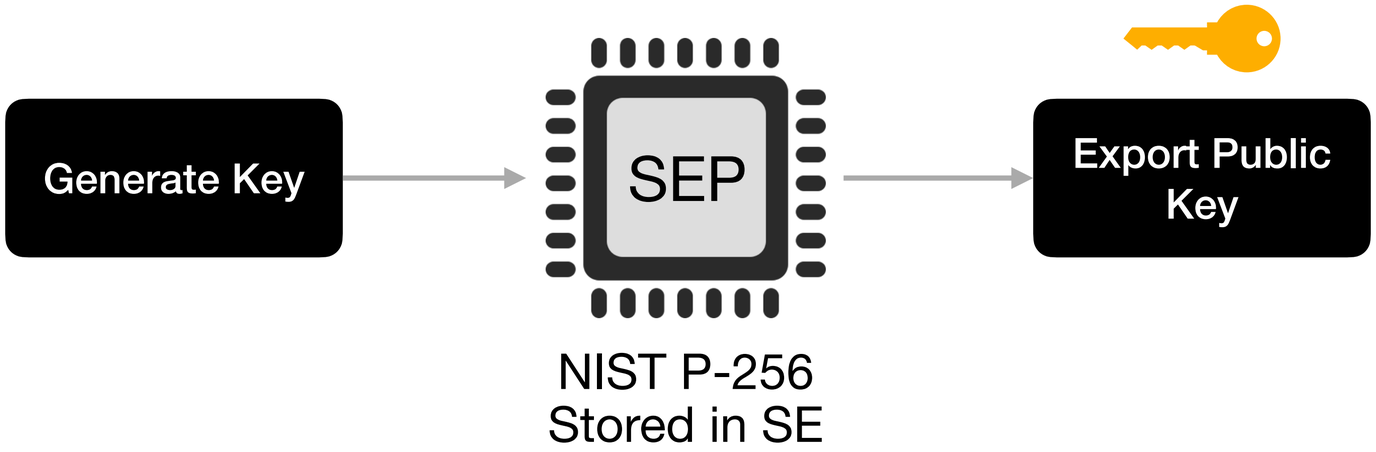


Language compression codecs used for text messaging during emergency communication.

LLC Keys



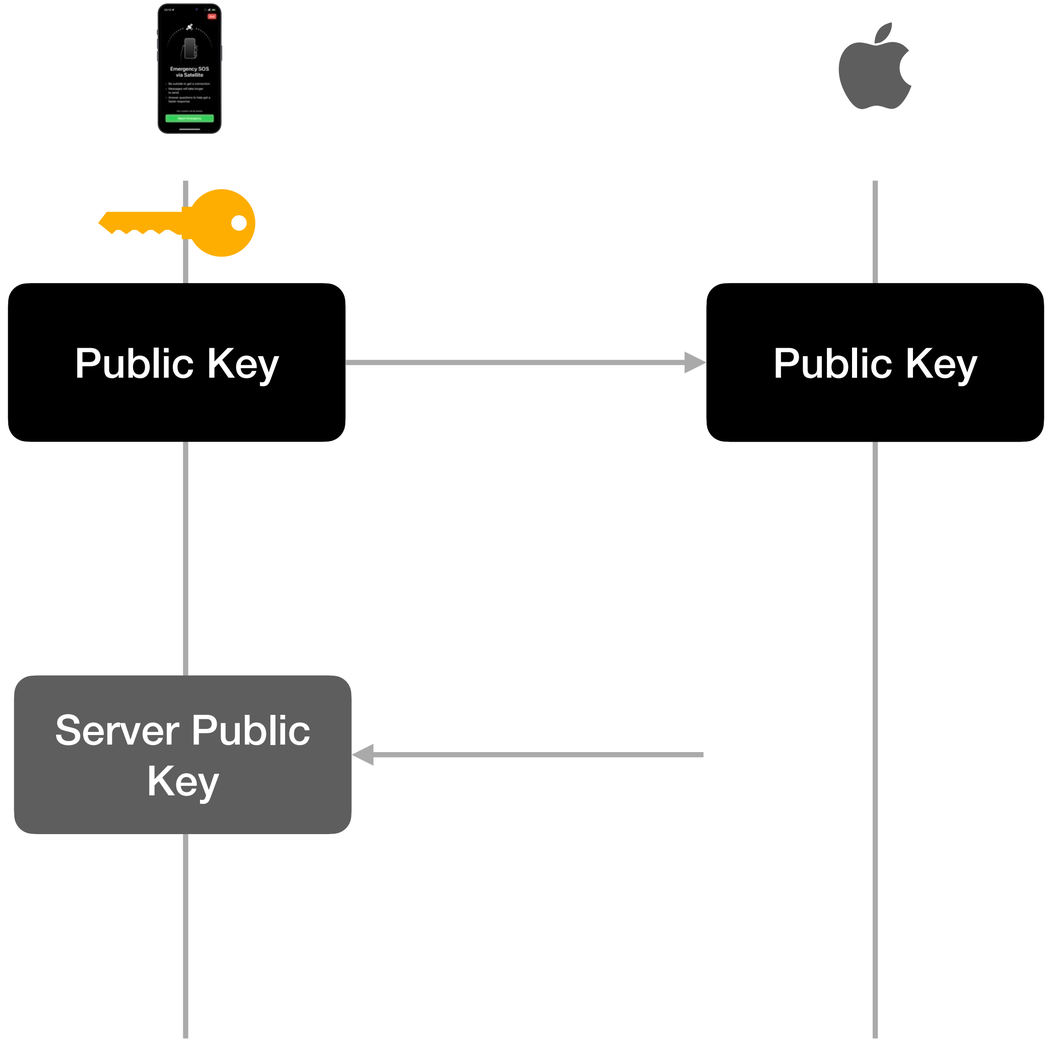
LLC Keys




```
var keyAttributes = NSMutableDictionary()
keyAttributes[kSecAttrLabel] = "com.apple.commcenter.llc"
keyAttributes[kSecAttrTokenID] = kSecAttrTokenIDSecureEnclave
keyAttributes[kSecClass] = kSecClassKey
keyAttributes[kSecAttrAccess] = kSecAttrAccessibleAlwaysThisDeviceOnly
keyAttributes[kSecAttrIsPermanent] = true
keyAttributes[kSecAttrKeySizeInBits] = 256
keyAttributes[kSecPrivateKeyAttrs] = kSecAttrKeyTypeECSECPrimeRandom

var error: NSError?
let success = SecKeyCreateRandomKey(keyAttributes, &error)
```

Key Synchronisation

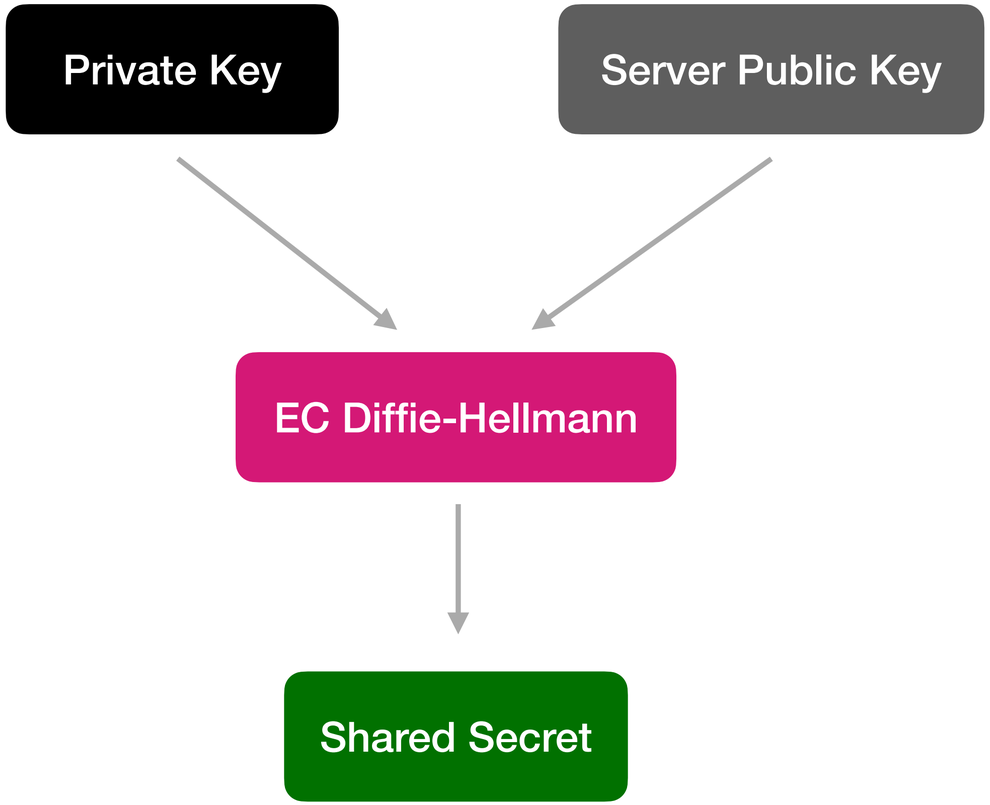


iOS Keychain

```
{
    agrp = apple;
    bsiz = 256;
    decr = 0;
    drve = 1;
    encr = 0;
    esiz = 0;
    extr = 0;
    kcls = 1;
    klbl =
    0xede4c2fa65c88fe965720d719ea80c5e8
    39195d8
    labl = "com.apple.commcenter.llc";
    perm = 1;
    priv = 1;
    sign = 1;
    sync = 0;
    sysb = 1;
    tkid = "com.apple.setoken";
}
```

```
{
    acct =
    "com.apple.commcenter.Stewie.STKData.
    ede4c2fa65c88fe965720d719ea80c5e839
    195d8";
    agrp = apple;
    labl = "com.apple.commcenter.llc.stk";
    svce = CommCenter;
    sync = 0;
    "v_Data" = {length = 109, bytes =
    0x62706c69 73743030 4f104104 4914d1f9
    ... 00000000 0000004c };
}
```

Session Key

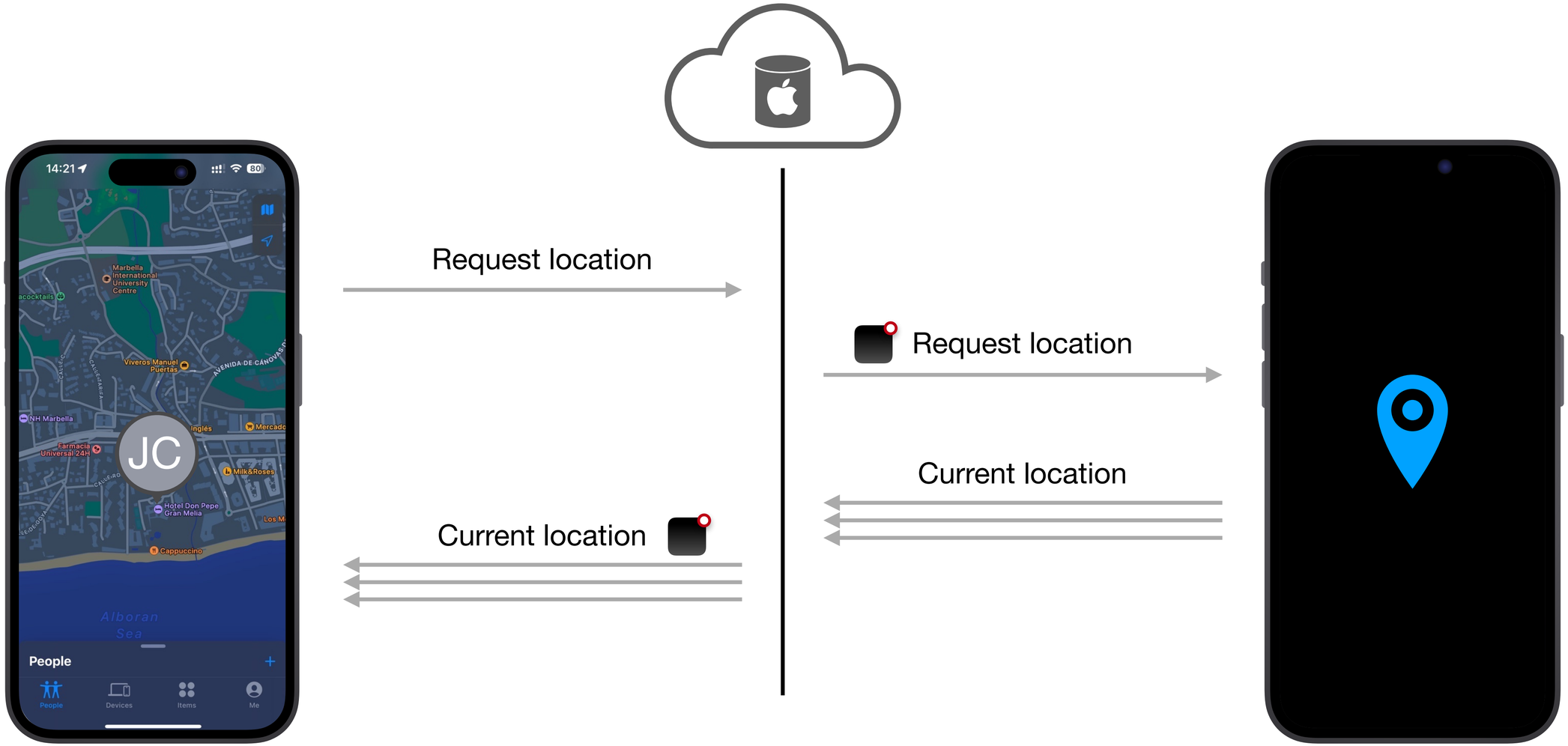


EPKI:%s, SECRET:%@

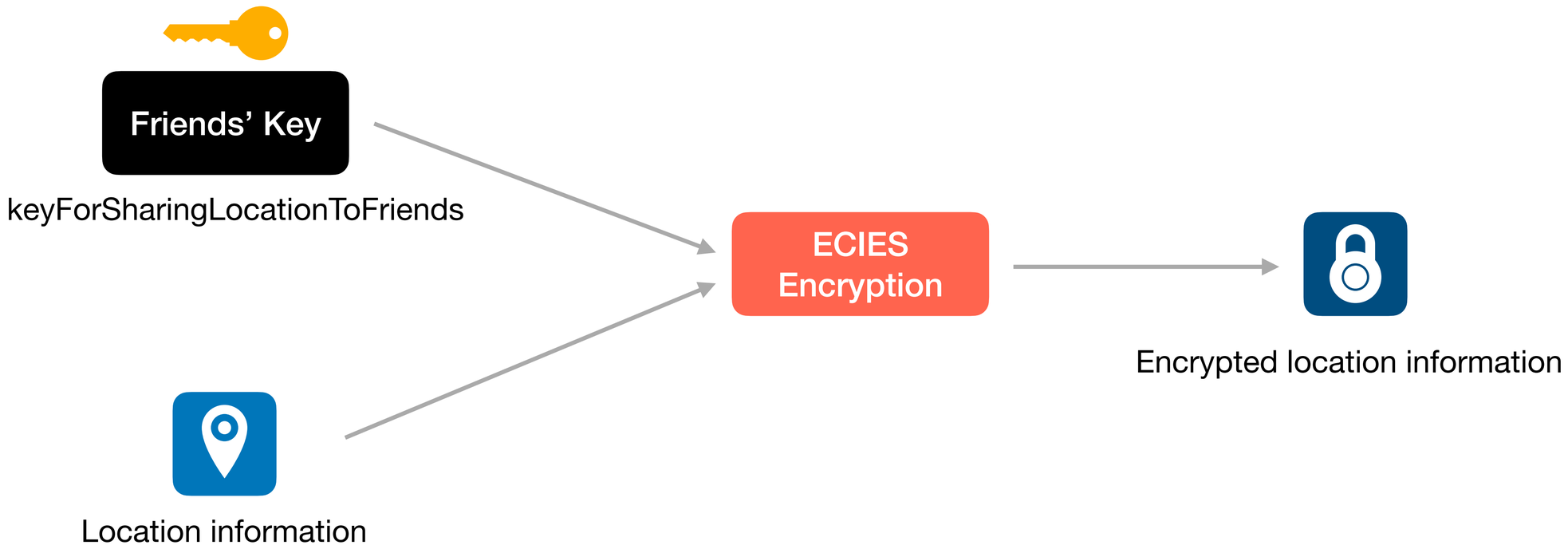
unable to generate shared secret for EPKI:%s and LBL [%@], error


Find My Location Sharing

Normal location sharing



End-to-End Encryption




```
eciesEncryptionStandardVariableIVX963SHA256AESGCM
```

Server Communication

Request


```
{
    "fetch": {
        "intent": "startLocationUpdates",
        "fmId": "NDMxNjY2NDIx",
        "mode": “live",
        "ids": [

"5832C58C5C48CEF1C27D3DF1552707FE42745B966AD75BA51D04B
7B34A698CD2"
        ]
    },
    "clientContext": {
        "apsToken":
"5832C58C5C48CEF1C27D3DF1552707FE42745B966AD75BA51D04B
7B34A698CD2",
        "contextApp": "com.apple.findmy.fmfcore",
        "clientId": "00008110-000260C02640011E",
        "shallowStats": {}
    }
}
```

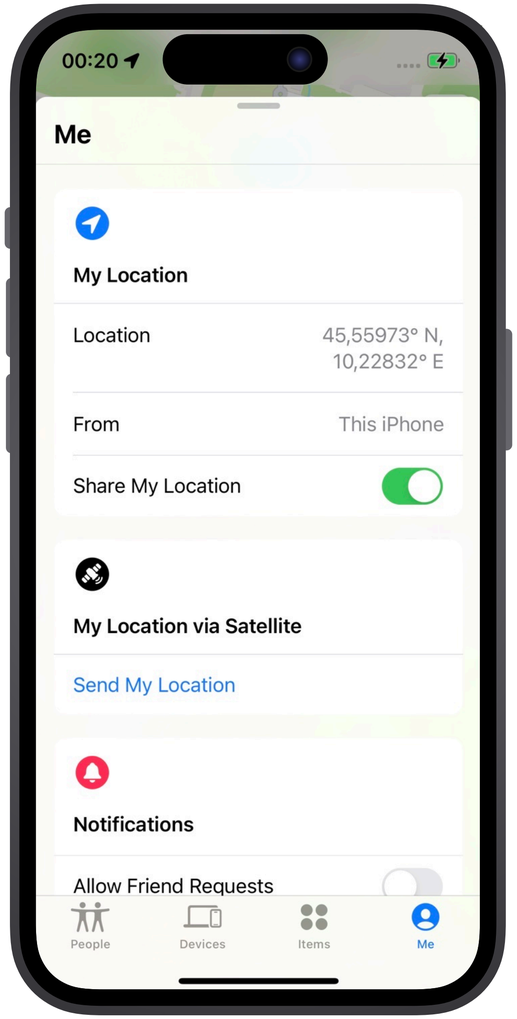

Response


```
{
  "locationPayload": [
    {
      "locationInfo": [
        {
          "locationTs": 1694186159999,
          "location":
"BFoX1TnSgzckVvD8mRxBGWLQFFJAKkTnzd0HAZZLCdvyan8M+E6GpiHSN6gQnmRV
n7IBY7DQslaBGRNUJPdU+4ReykOKDqYg8O7DrIVgb+OWLwgH1IYBtGIUv8CdlGHn3
FxePeP1LJYn4vEDQoA2zUervtNEfsM6zU4McUipF1aHg4x//
sKU6mxekrQ+r+OMiBkfA0yAyGKbjfaLv+fy9ZNLnuFKVAtjqywLDgedZK4q43RbYD
ksdaZ0Bvlp6jql7qoaoUZfH1owVWccm9aRcrZtARa/j/UIRR3ClhQSYCRutq/
gubxqo27gy+5wluhM7GLB5dwn0KPJkP+kVWNOj5EvXAPxQaSnoYicIOwLhgDhwj/
wrZaqYnsr9Listjh/
5j+DAvtKVZARD4x+RDvdhabiF1Y9vrX2SG9+3KX+QXrjQ+6k+vtm6mAp+ihEj/
lABAHknYkwZkT88m4CGK48pDLkCCh/
1A3EARimdVAcUN0Yj7AbzPEb9PiTA2+v8eEtWPrKtY6O6w48P4Ul",
          "fmt": 0
        }
      ],
      "id": "9CxlEA8ublwnacqnTQAsdsAp4g0cmp13nipiEAVniuOo="
    }
  ],
  "configVersion": 21,
  "statusCode": "200"
}
```

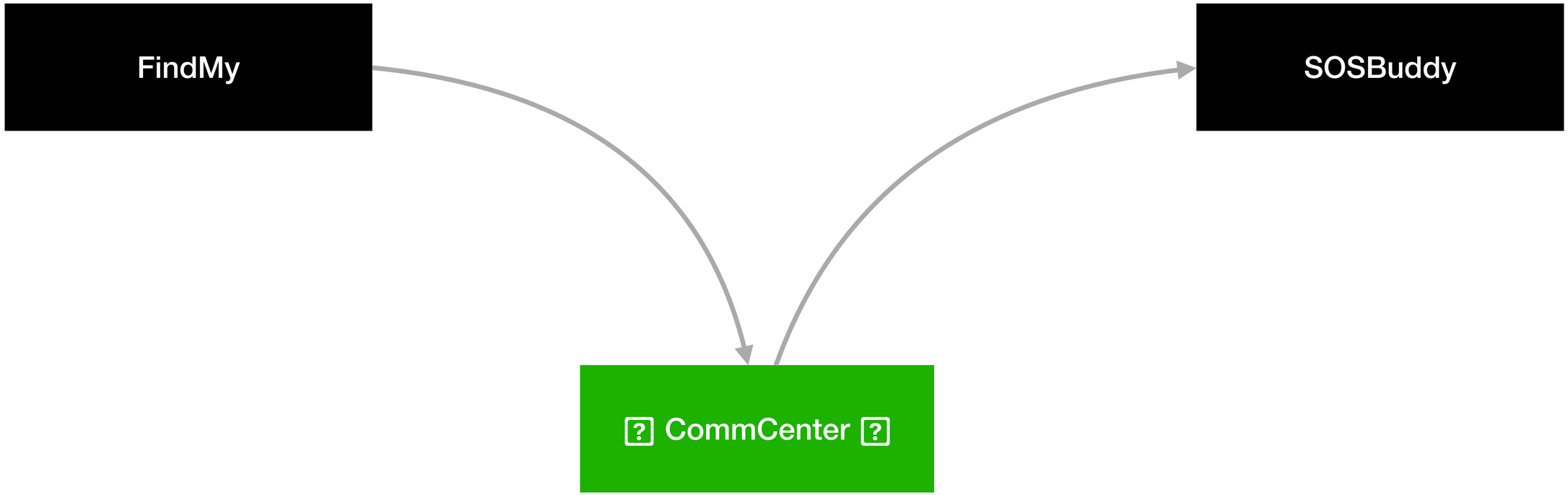

Satellite Location Sharing

Involved Processes

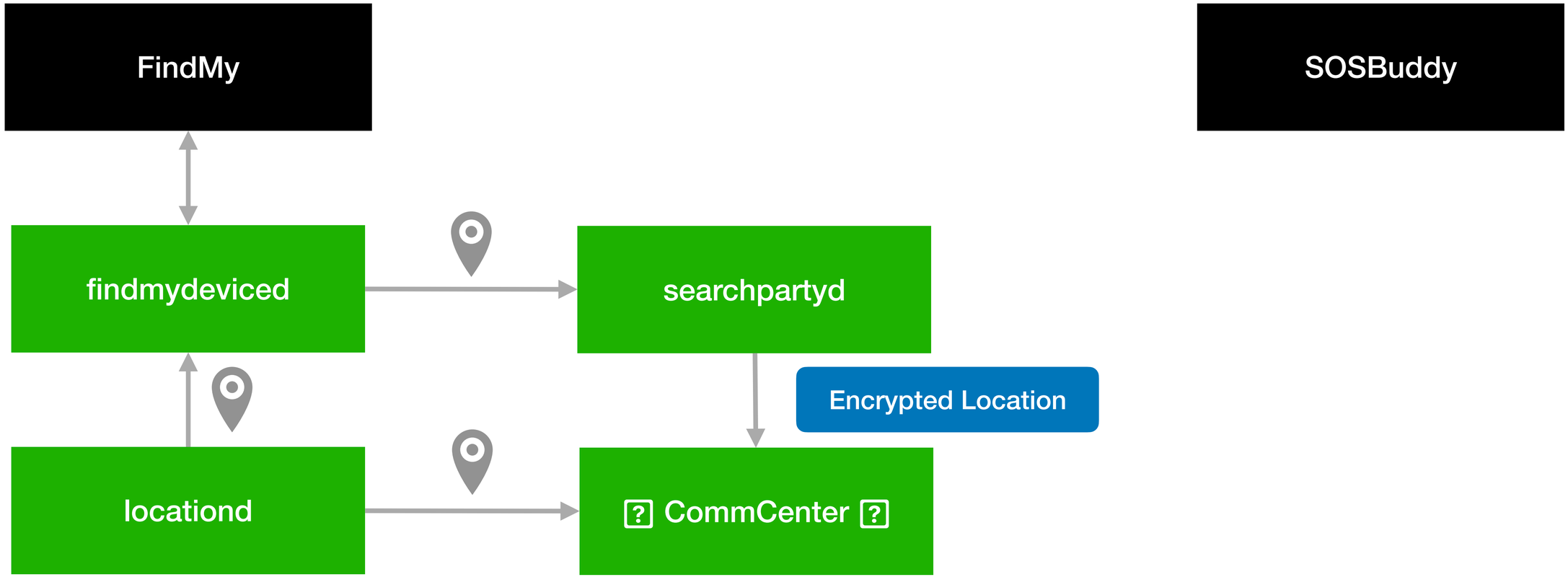
FindMy



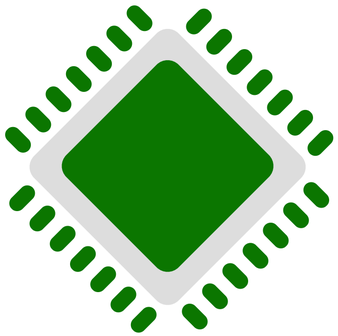
Involved Processes



Involved Processes



Involved Processes



Qualcomm Baseband

CTStewieTransportMessage

CTStewieFindMyMessage

Encrypted Location

CommCenter

Satellite Transmission

Reducing Data Size

Lite
Location

Satellite Transmission

```
Int32(latitude * 10,000,000)
+
Int32(longitude * 10,000,000)
+
UInt8(horizontalAccurazy)
=
```

Lite Location 9 bytes

Satellite Transmission

```
Int32(latitude * 10,000,000)
                +
Int32(longitude * 10,000,000)
                +
 UInt8(horizontalAccurazy)
                =
```

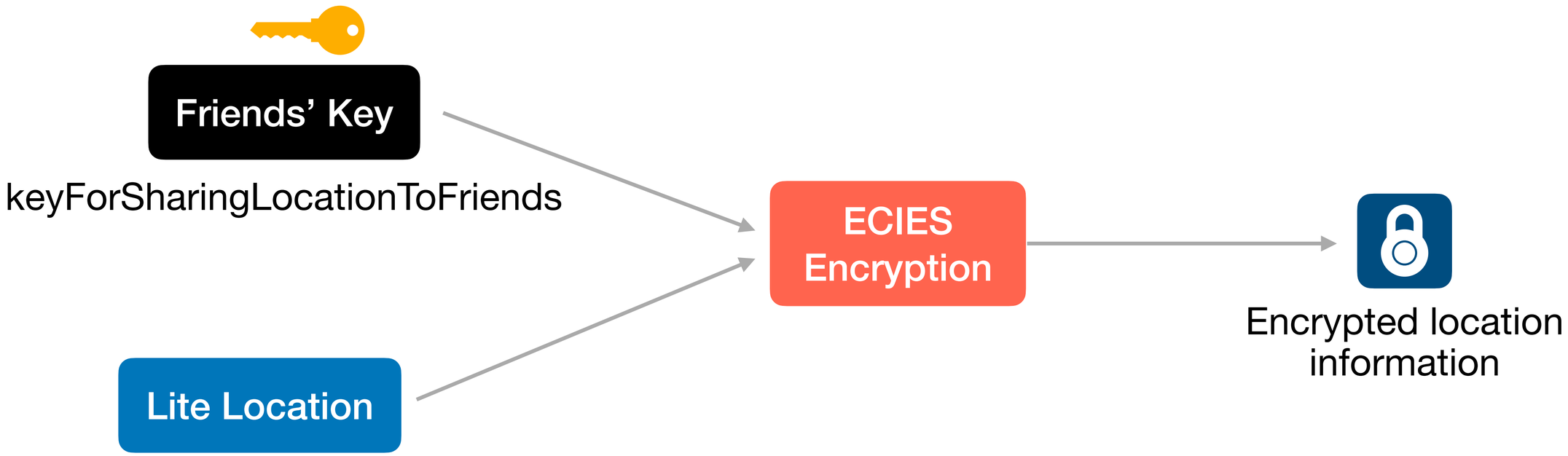
Lite Location — 9 bytes

48.431139 08.656773

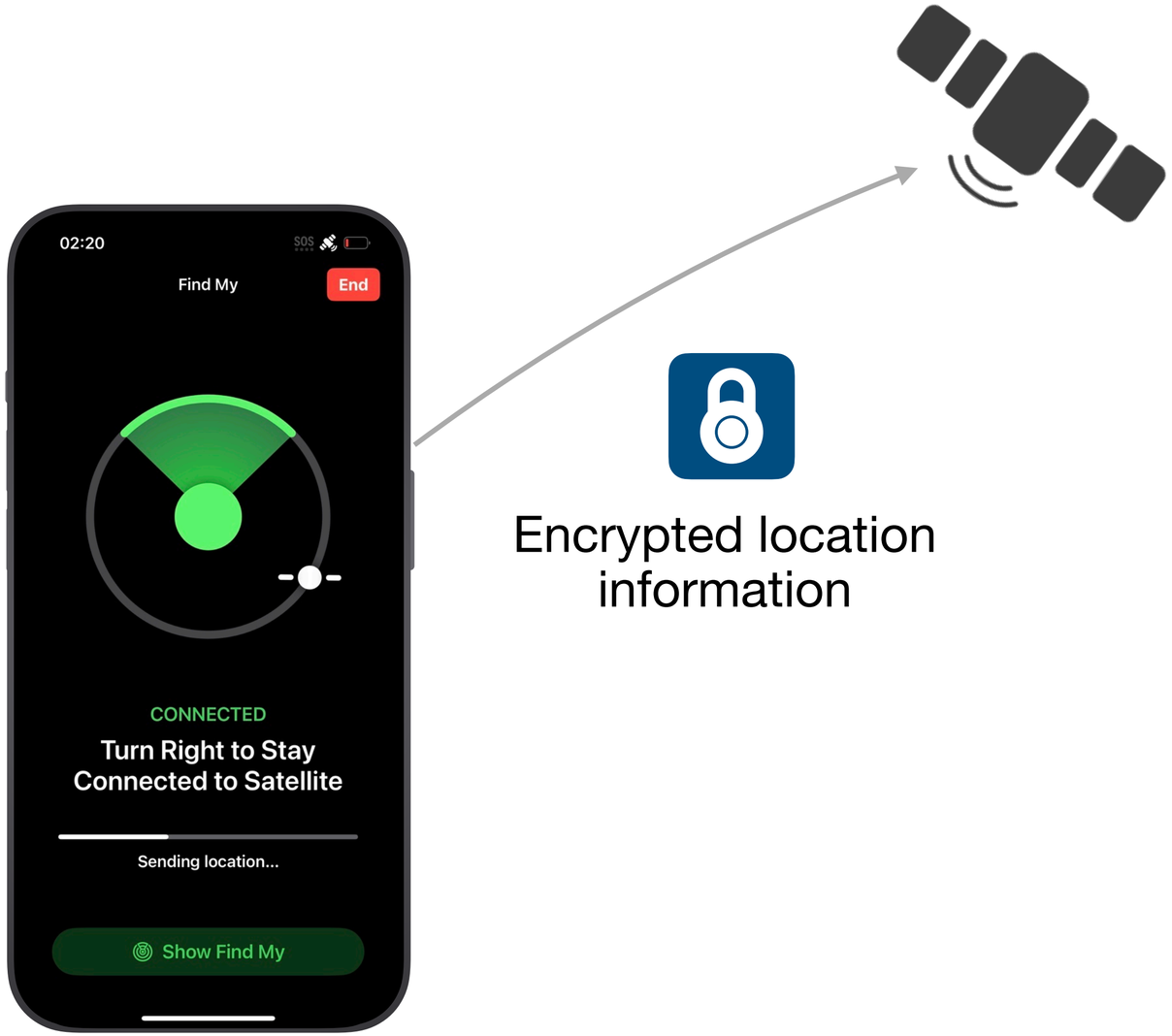
4843113900 865677300


AirTag

Satellite Transmission



Satellite Transmission



Misuse

Misuse of Satellite Communication

Sending Messages for Free

- Satellite messages are expensive
- Location sharing can only be used every 15 minutes
- Bandwidth is limited
- Millions of users

Manipulating the Location

...to send messages

Latitude : -90.0 - 90.0

Longitude: 0.0 - 180.0

Manipulating the Location

...to send messages

Latitude : -90.0 - 90.0

Longitude: 0.0 - 180.0

8 bytes of data

Mapping the Alphabet

...on GPS Coordinates

A-Z a-z 0-9

62 chars

`10,000,000,000`

10 digits space

`2 digits = 1 char`

5 chars per coordinate

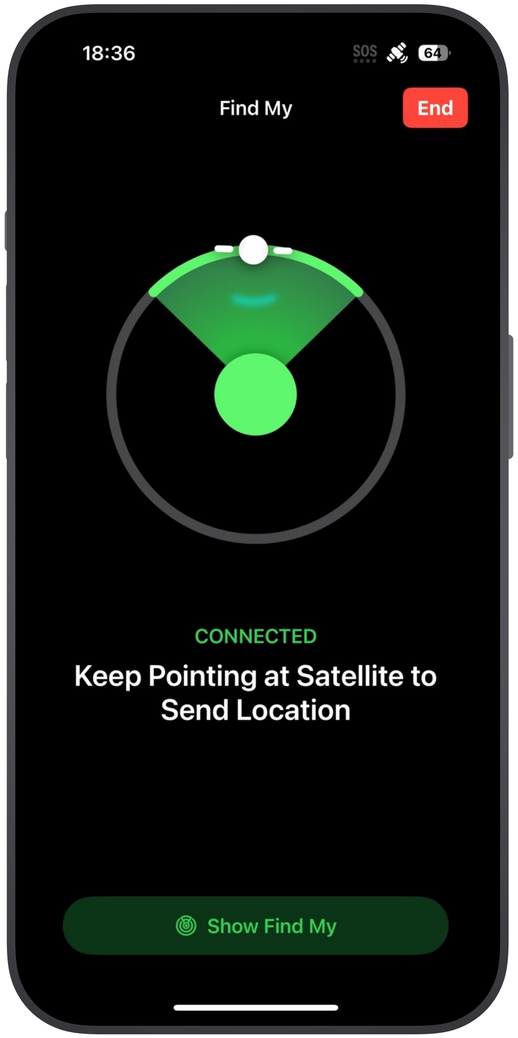
`HelloWorld`

10 characters per message

Apple was smarter!

CommCenter has checks if location source was simulated

😩



🤔

ios 16 jelbrek for iphone 14, wen eta?

Jiska Classen

Group Leader

Hasso Plattner Institute

jiska.classen@hpi.de

Alexander Heinrich

Security Researcher & PhD Candidate

Technical University of Darmstadt

aheinrich@seemoo.de

Image Sources

- Slide 2,4,8,12,14,42,52: Generated by Midjourney
- Slide 9: Pixabay
- 11: findmespot.com (official product image)
- Satellites, iPhone frames, processor chips are self drawn