Remote Access to IoT Devices: Common Needs and Approaches

Eystein Måløy Stenberg

December 2nd, 2021

About me

- Eystein Stenberg
  - Co-founder of Mender.io
  - 10 years in systems security management
  - M. Sc., Computer Science, Cryptography
  - eystein@mender.io
- Mender.io
  - Remote software management for connected devices (OTA updates)
  - Open source core (ASLv2)
  - Add-ons for device management use cases

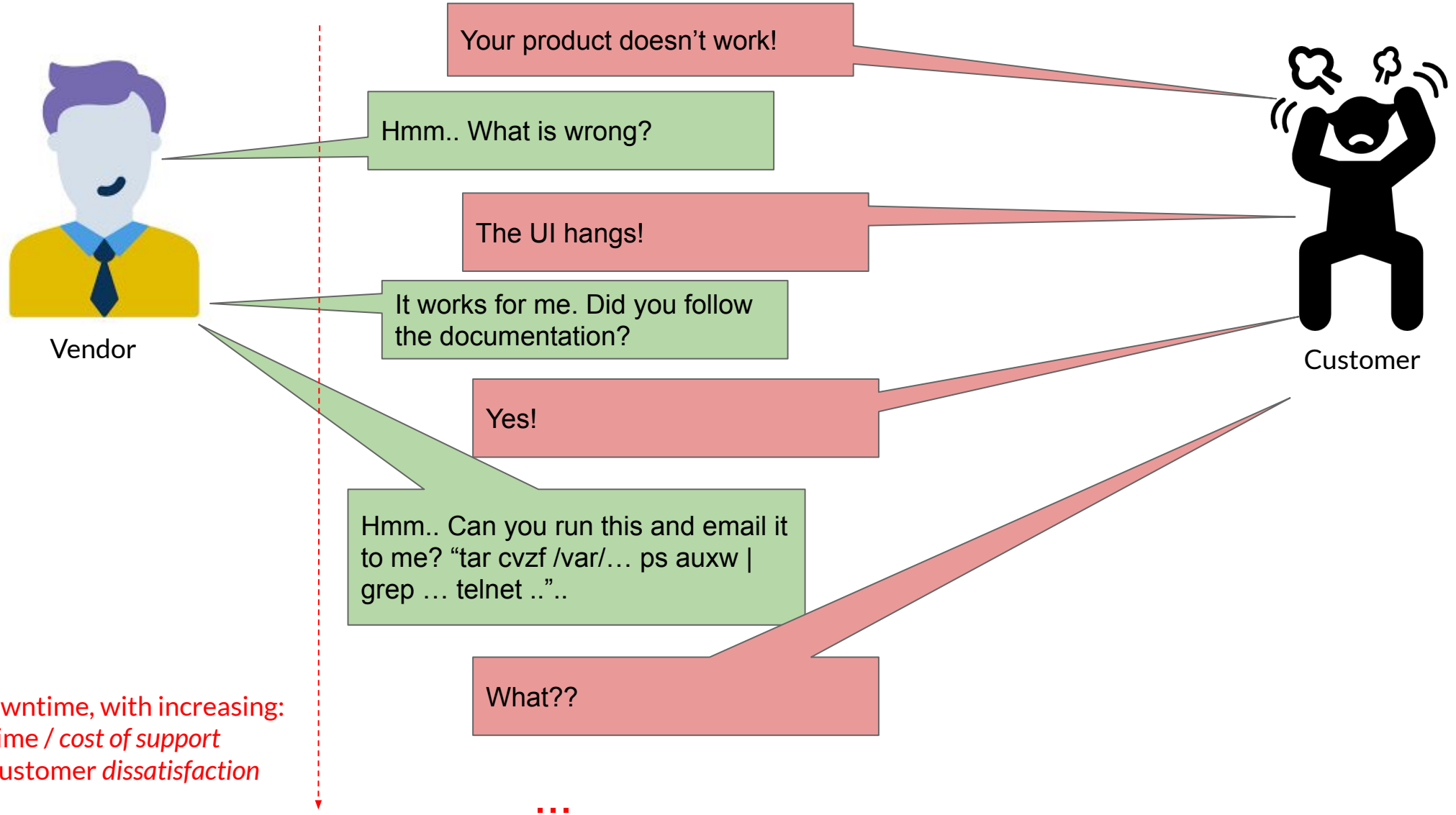
Motivation

- Remote access is a common need in IoT
- Mender recently researched user needs and technology
  - 30+ interviews + surveys
- Share findings to save you time and pain in the future

Agenda

- **What is Remote access ?**
- Requirements for Remote access in IoT
- Solutions and comparison of technologies

Has this ever happened to you?



Downtime, with increasing:
- Time / *cost of support*
- Customer *dissatisfaction*

Remote access is used to

- *Troubleshoot & hotfix issues* surfacing from:
  - Customer support
  - Monitoring / Alerts
- Make ad-hoc changes to development / pilot environments
- Typically about a *single device* at the time
  - You already know which device needs attention

Remote access is part of Device Management

OTA/Software updates

"Quickly and safely improve product."

Configure

"*Customize* each device to its environment."

Remote access

"Resolve [support] issues real-time, in a secure way."

Monitor

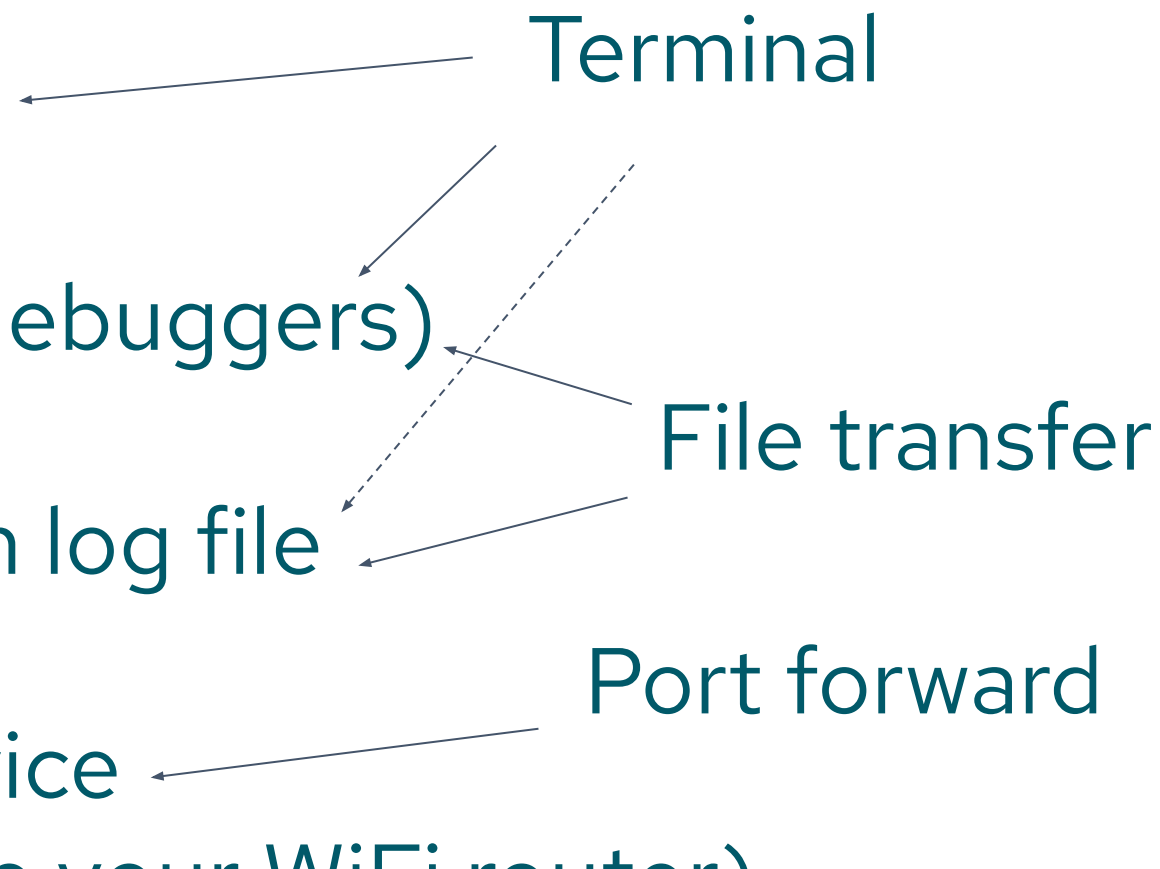
"Detect and analyze *health issues* of devices, services and applications."

Agenda

- What is Remote access ?
- **Requirements for Remote access in IoT**
- Solutions and comparison of technologies

Typical Remote access use cases

- Restart application / device
- Run diagnostics tools (e.g. debuggers)
- Analyze application / system log file
- Access local services on device
  - “Device admin portal” (like your WiFi router)
  - Test connectivity / responses as seen from device



Remote access requirements: Use cases

1. Terminal

2. File transfer
   a. Bidirectionally

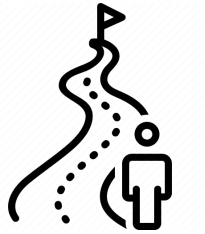
3. Port forward

The analogy to **ssh** and **scp** in cloud infrastructure

- Well understood use cases, very feature rich

- Cover all the use cases:
  - Terminal: ssh
  - File transfer: scp
  - Port forward: ssh

The IoT environment



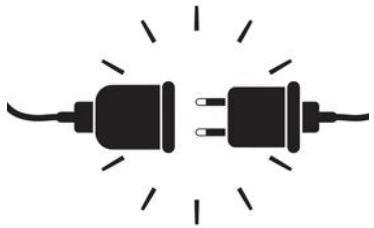
Remote
- Expensive to reach physically

Long expected lifetime
- 5 - 10 years

Remote access becomes more important

Unreliable network
- Intermittent connectivity
- Only outbound connections
- Insecure
- Low bandwidth

Remote access becomes difficult



Unreliable power
- Battery
- Suddenly unplugged

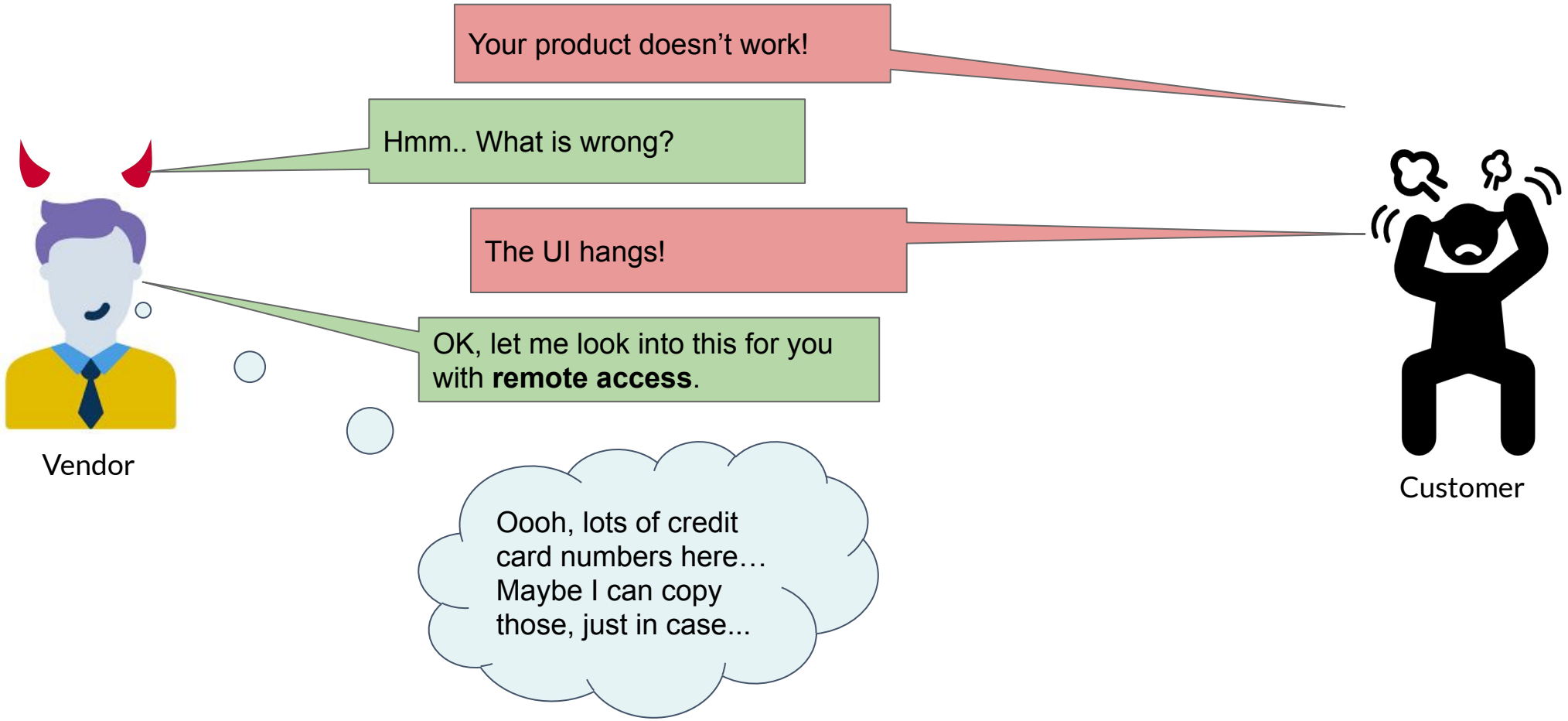
Remote access requirements: Network

1. Only outbound connections (from device)
   - (for connectivity)

2. Secure end-to-end
   - Authenticated and encrypted
   - Bi-directionally
   - Zero open ports on device

3. Low network overhead

One more problem: Remote access grants wide control



Remote access requirements: Operator security

1. Audit logs
   - Who did what, when & to which device?
   - Terminal session log

2. Approval of access on demand (not “always on”)

3. Role Based Access Control (not “all users, all devices”)

4. Device-side user restrictions (not “root”)

Summary of requirements for Remote access in IoT

- Terminal
- File transfer
- Port forward

- Outbound connections
- End-to-end secure
- Low bandwidth

- Audit logs
- Access approval
- RBAC
- Device-side restrictions

Agenda

- What is Remote access ?
- Requirements for Remote access in IoT
- **Solutions and comparison of technologies**

Narrow the options: Two most restrictive requirements

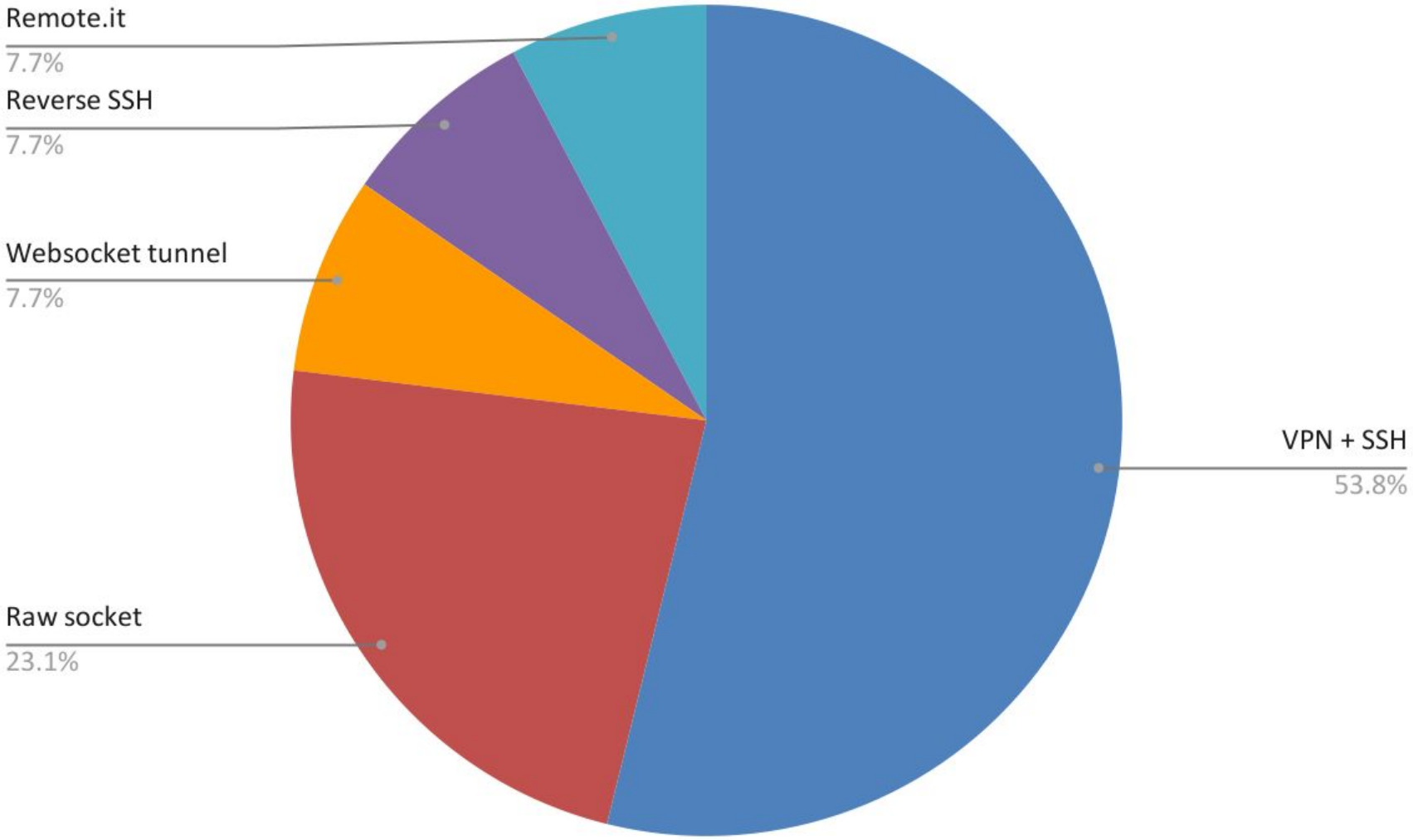
- Terminal
- File transfer
- Port forward

- Outbound connections
- End-to-end secure
- Low bandwidth

- Audit logs
- Access approval
- RBAC
- Device-side restrictions

How to access the **terminal** on a device having only **outbound** connections?

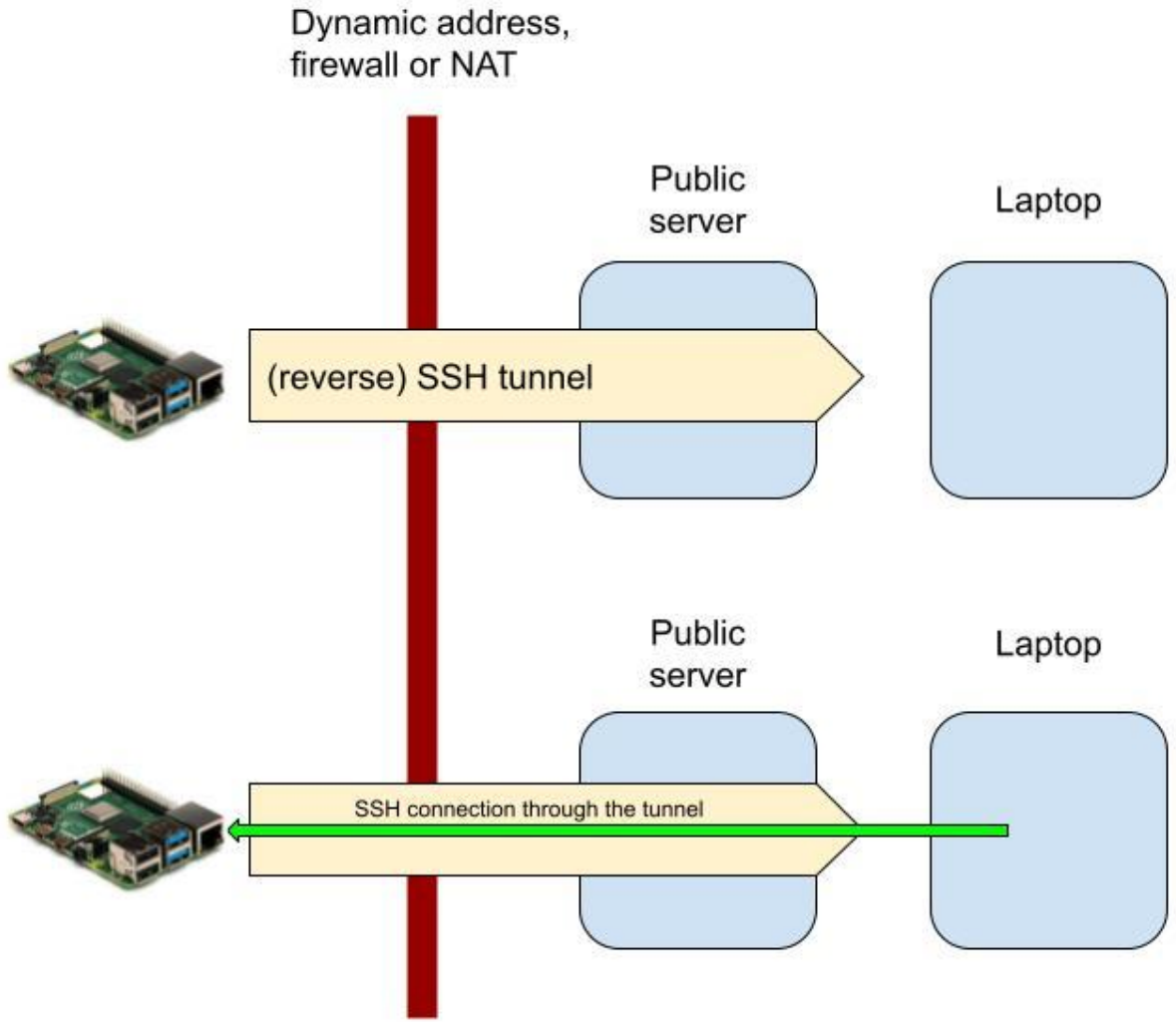
User interviews - how is terminal access provided today?



Option 1: VPN

- Allows for inbound connections on overlay network
  - I.e. eliminates the “outbound connections” requirement
- Used heavily in cloud & desktop environments
- Several open source implementations
  - Wireguard
  - OpenVPN

Option 2: Reverse SSH

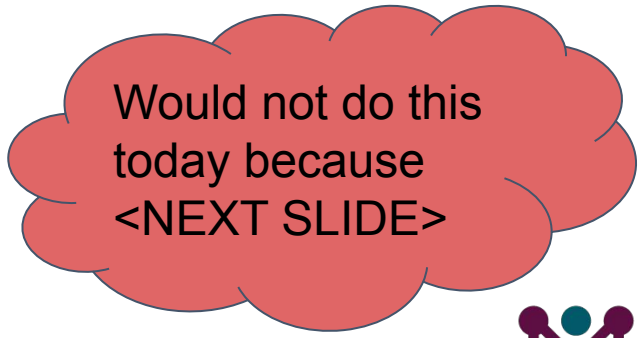


- Exposes a port on a public server for access to a specific device
- Compared to VPN
  - Easier to set up
  - Harder to maintain

Would only use
in PoC / Pilot

Option 3: Raw socket

- Persistent TCP connection initiated from device
- Server side keeps connections alive and lets users use them as needed for terminal access
- Must go through a secure channel
  - TLS
- Custom, homegrown implementation



Option 4: WebSocket

- Specified in 2011 (IETF)
  - Wide client & server support

- Built on HTTPs (“Connection upgrade”)
  - Bi-directional & full duplex
  - Traverses firewalls more easily
  - Support for HTTP proxy & reverse proxies

- Takes care of connection management
  - Health check
  - Keepalive

Option 5: Off-the-shelf solutions

- Generally good use case coverage
- ...typically covers *more* than these use cases as well
- Some examples
  - Remote.it (Proprietary SaaS, $72 / user year)
  - Mender Troubleshoot add-on package (Open source core)
  - Many more...

What about MQTT?

- From mqtt.org:
  - “MQTT is an OASIS standard **messaging protocol** for the Internet of Things (IoT).”
  - “It is designed as an extremely lightweight **publish/subscribe messaging** transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth.”

- MQTT is a *messaging and pub/sub protocol*, *WebSocket is a transport*.
  - Do you need pub/sub for Terminal?

Now, take a step back...

- Terminal
- File transfer
- Port forward

- Outbound connections
- End-to-end secure
- Low bandwidth

What about the rest of the requirements?

- Audit logs
- Access approval
- RBAC
- Device-side restrictions

What are the remaining options?

1. VPN
2. ~~Reverse SSH~~
3. ~~Raw socket~~
4. Websocket
5. Off-the-shelf

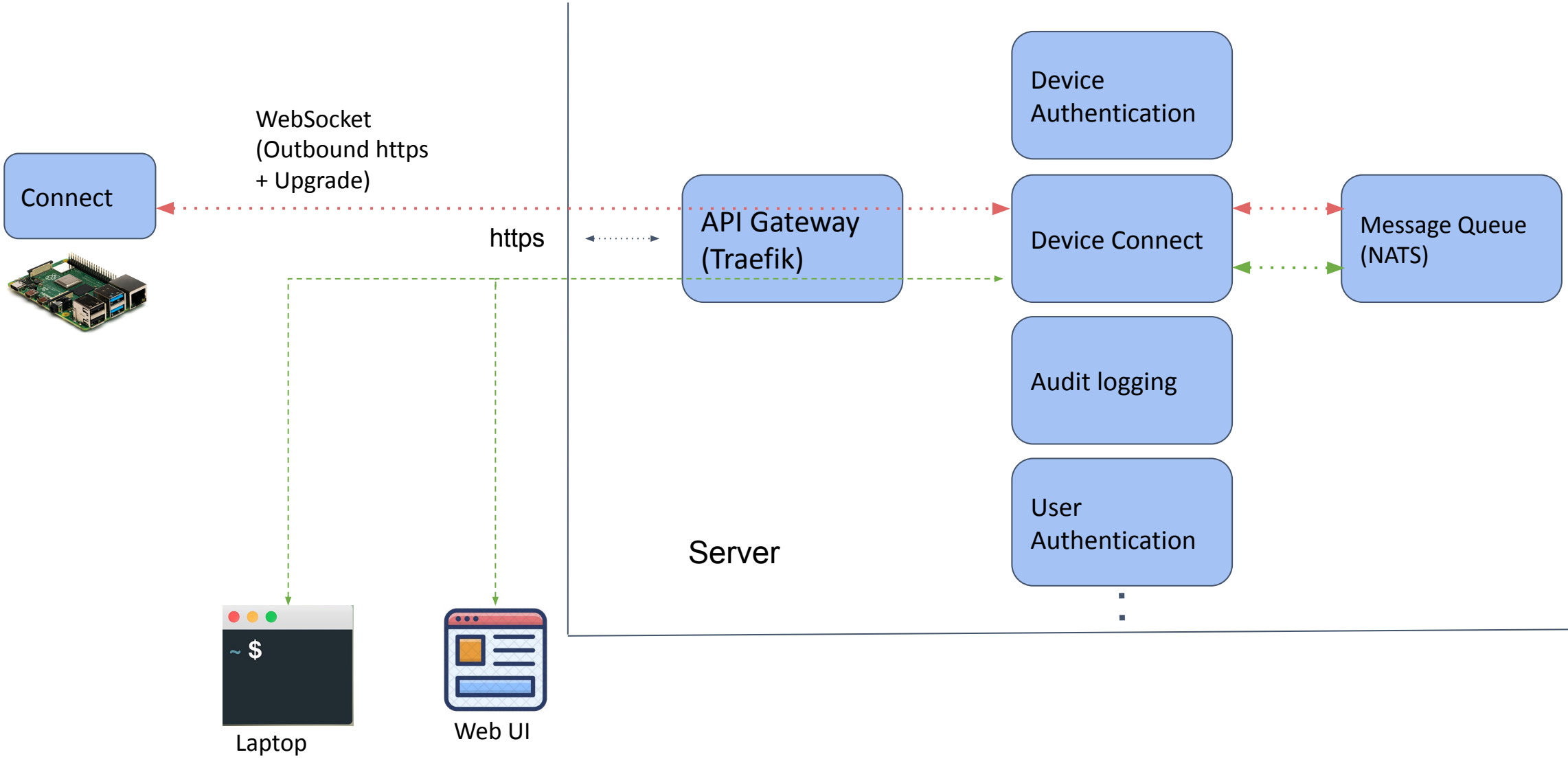
VPN + SSH

Terminal	✅	
File Transfer	✅	
Port forward	✅	
Outbound connections	✅	
End-to-end secure	⚠️	Open port on device, more credential mng.
Low bandwidth	⚠️	Tunnel in tunnel (+higher infra complexity)
Audit logs	❌	Can be built on top, but need to auth. *user*.
Access approval	❌	
RBAC	❌	Very difficult to build on top
Device-side restrictions	✅	Only use restricted user account (AllowUsers)

WebSocket based (used in Mender Troubleshoot)

Terminal	✅	
File Transfer	✅	
Port forward	✅	
Outbound connections	✅	
End-to-end secure	✅	Zero open ports, using Mender OTA creds.
Low bandwidth	✅	Simple character piping via WebSocket
Audit logs	⚠️	Available in commercial version
Access approval	❌	
RBAC	⚠️	Available in commercial version
Device-side restrictions	✅	Can configure which user to use on device

Example WebSocket-based architecture (used in Mender)



Do you have infrastructure on you device you can reuse?

- VPN connection?
  - Can add SSHd
  - NB! Open ports (to where?) and credentials used (shared?).
- MQTT (over TLS)?
  - Can build Terminal on top (not sure about Port forwarding)
- TLS keys?
  - Websocket, or off-the-shelf solution
- Nothing?
  - Off-the-shelf solution

Reuse an existing solution for Remote access

- Building & maintaining Remote access takes more time than you think
- Consider infrastructure already in place on your devices
- Consider an off-the-shelf solution *built for IoT*
- Focus on developing your product instead

Thank You

Q & A