An Overview of the User Services Platform (USP) (Broadband Forum TR-369)

Prepared especially for technical people
- most meaningless marketing terms have been removed

What we’re going to talk about

- TR-069 (CWMP) history lesson and where it’s lacking (feel free to skip slides 3-6 or go through them quickly if you don’t care)
- High level User Services Platform (USP) (TR-369) use cases (slides 7-11)
- How it works – a look at the underlying technologies and features of USP (slides 12-14)
- Driving interoperability and deployment (slide 15)
- USP Resources (specification, data model, etc.) (slides 16-17)

LET’S GO BACK IN TIME

In early 2000s, **broadband gateways** became a regular part of operator deployments.

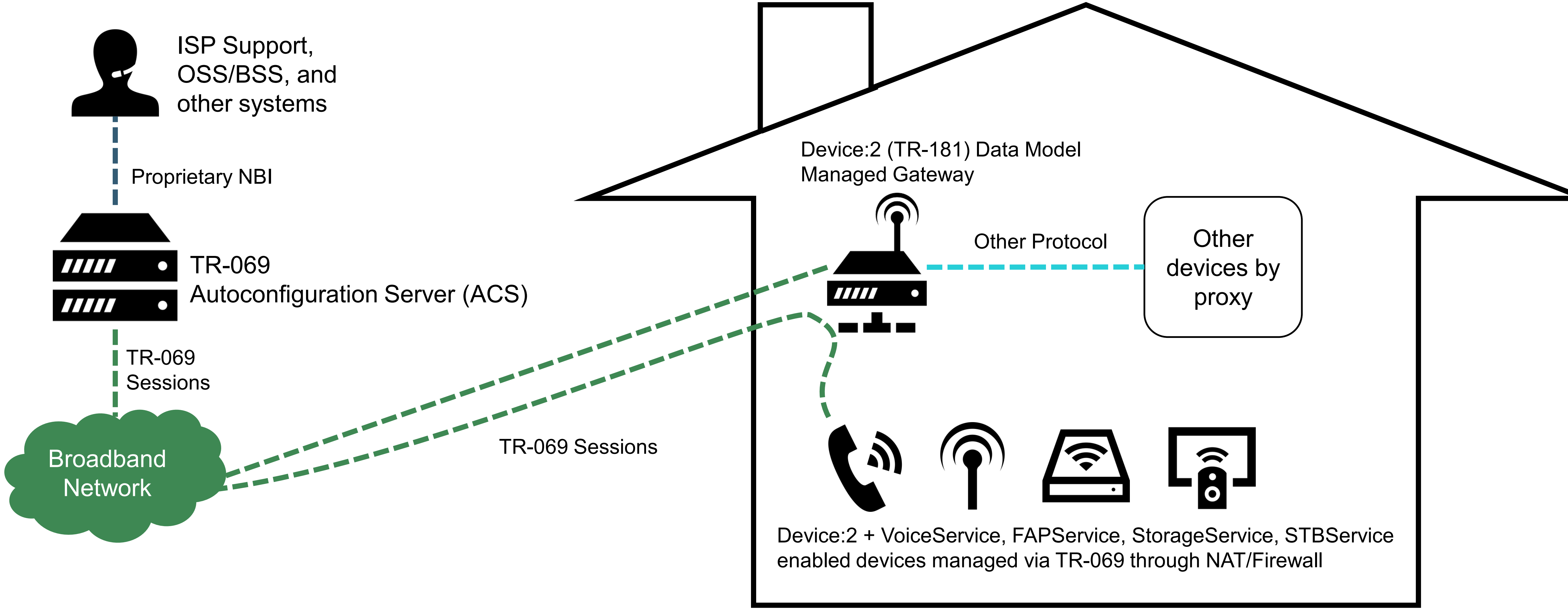
Deploying, **onboarding and managing the broadband gateway was hard!** Truck rolls, CD-ROMs, UPnP…

The key issues: **LIFECYCLE MANAGEMENT, MAINTENANCE AND MONITORING, PROVISIONING NEW SERVICES**

TR-069 CPE WAN Management Protocol (CWMP)

TR-069 Architecture

Single ACS operated by ISP manages devices with a standardized data model over HTTP

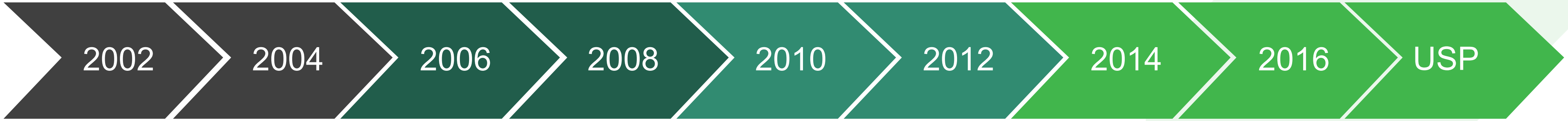


ISP Support, OSS/BSS, and other systems
Proprietary NBI
TR-069 Autoconfiguration Server (ACS)
TR-069 Sessions
Broadband Network
TR-069 Sessions
Device:2 (TR-181) Data Model Managed Gateway
Other Protocol
Other devices by proxy
Device:2 + VoiceService, FAPService, StorageService, STBService enabled devices managed via TR-069 through NAT/Firewall

The evolution of managed user experience

ISPs see **need for life-cycle management, monitoring, and provisioning** for gateway routers. CWMP (TR-069) is born.

Cable/MSOs incorporate TR-069 for management of **advanced gateways/Wi-Fi using Device:2** data model.

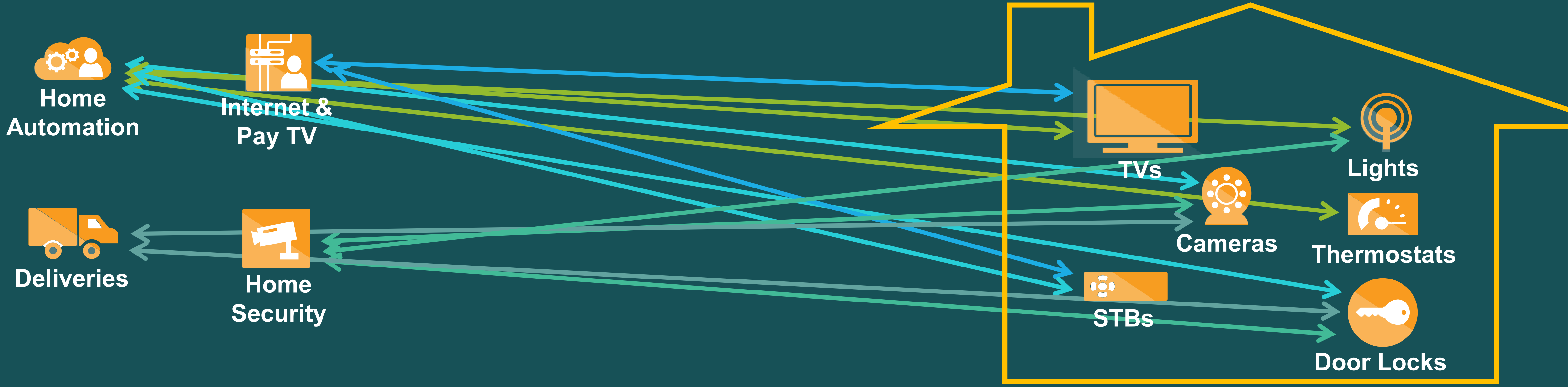


2002 2004 2006 2008 2010 2012 2014 2016 USP

TR-069 **expands to manage more** interfaces and more devices such as: STB, VoIP, Wi-Fi, and more.

Explosion of new technologies and challenges for both networking and consumer electronics: IoT, Wi-Fi/Mesh, handling over-the-top and third party services, and desire for end-user control.

Why is this evolution necessary?



Bigger SCOPE

- New devices, new services, and the presence of virtualization
- User control, enabling 3rd party interactions
- Desire for seamless user experience, anywhere

Bigger SCALE

- Orders of magnitude more devices and connections
- More data/bulk telemetry needed to enable Machine Learning (ML)
- Increased need for real-time configuration management

Bigger STAKES

- Security and product lifecycle/upgrade concerns
- Privacy and data security concerns
- Ownership, responsibility and access control concerns

USP Use Case:
Explosion of Managed Devices

Problem: Between Wi-Fi Mesh solutions and Smart Home solutions, we are looking at an order of magnitude more devices in the connected home, all of which need to be remotely controlled and managed – in real-time and using mechanisms that scale.

USP Solution:

- Always-on communications reduces the number of messages sent across the network.
- Binary data encoding and relative path usage reduce the size of the messages sent across the network.

USP Use Case: App-Based End-User Management / Control

Problem: If it is in the house, we want an App to manage it: Wi-Fi… want that App, Smart Home… want that App, TV/Video… want that App. Not only do we want those Apps, they have to be responsive and consistently display relevant information.

USP Solution:

- Always-on communications leads to a more responsive experience.
- Using CoAP in the home network allows for resiliency.
- Robust and forgiving messaging allows for relevant information to be consistently retrieved despite variations in supported data models or home network conditions.
- Role-based authorization allows end users access to different functions than operators.

USP Use Case: Security and Privacy

Problem: We live in a scary new world where hackers are more than ready to take advantage of every attack vector to steal vital personal information. Our solutions need to be more secure than ever so we can protect everyone’s privacy.

USP Solution:

- Encrypting the communications channel via TLS/DTLS
- In cases where there may be untrusted proxies or brokers in the middle, messages need to be secured from the Controller to the Agent and back via an end-to-end message-layer security solution.
- Regular firmware upgrades ensure that attack vectors are closed before they are a problem.
- Strict access control rules ensure rogue Controllers can’t reach data that they aren’t allowed to access, and the line between operator data and user data can be made clear.

USP Use Case: Mass Telemetry

Problem: Operators want to understand the quality of experience being provided to their subscribers as it costs more money to get a new customer than it takes to keep an existing customer.

USP Solution: Efficient data collection that can optionally be decoupled from the remote management channel is the first step. Once the data has been collected, the Operators then have the opportunity to analyze and act on that data with the help of big data analysis tools and machine learning solutions.

USP Use Case

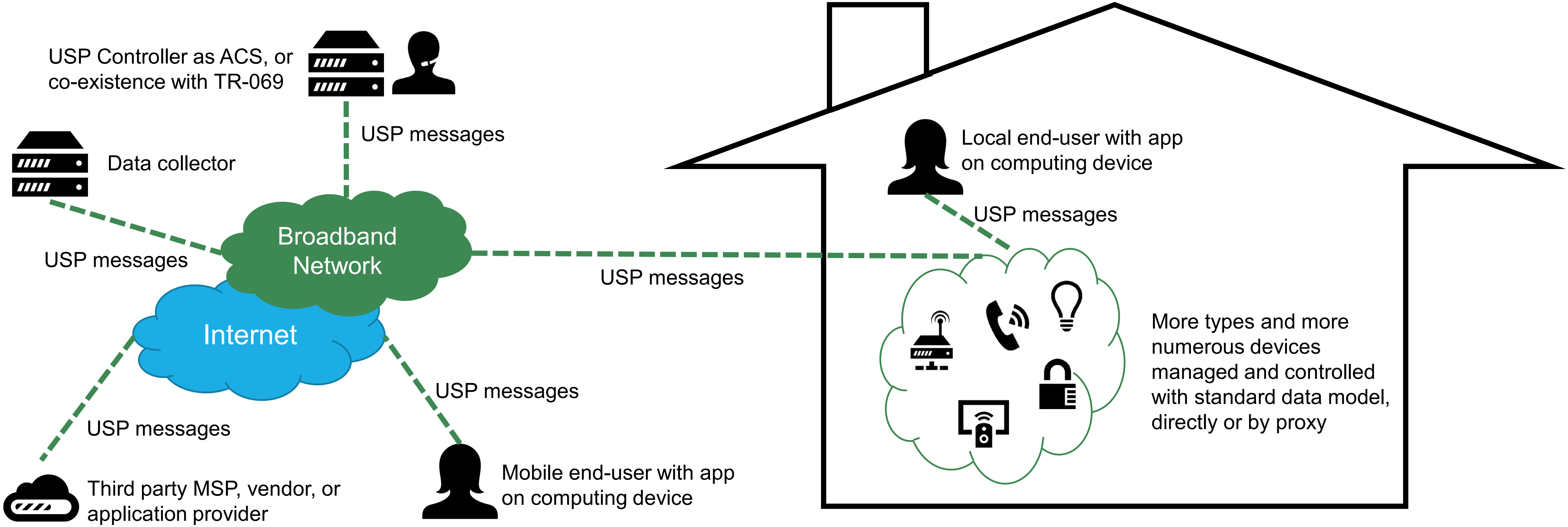
Greenfield vs. Brownfield

TR-069
+
USP

Problem: As with any new solution, you have to consider both Greenfield environments (new unconstrained environments) and Brownfield environments (existing environments with constraints).

USP Solution: Utilizing the Device:2 root data model was of key importance as it allows for backwards-compatibility and speeds up the time-to-market for new solutions. That doesn’t mean TR-069 CWMP deployments can be ignored as TR-069-enabled devices will be around for years to come… so co-existence between USP and CWMP is also very important.

USP (TR-369) Architecture

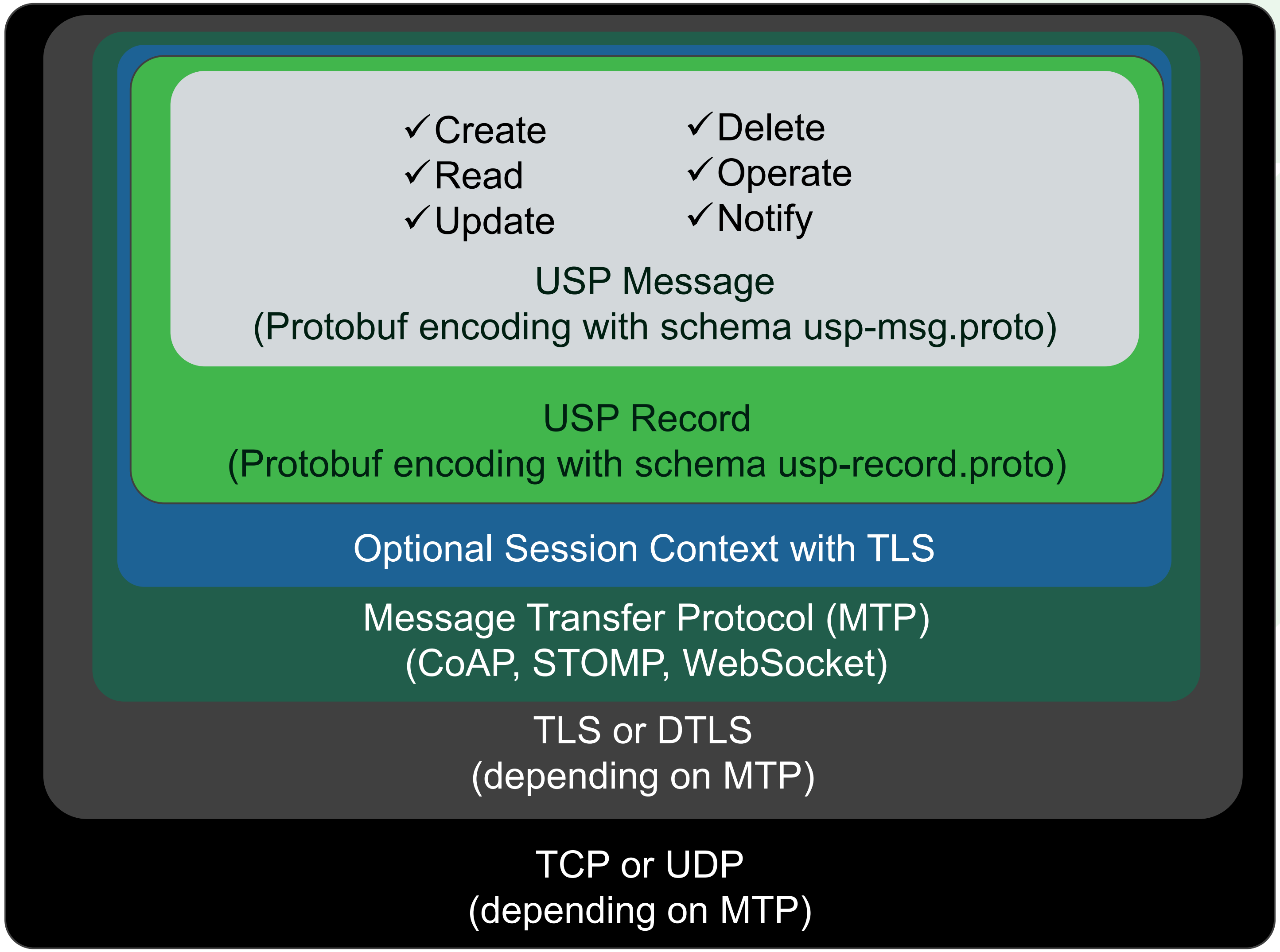


- Multiple USP Controllers can be anywhere in relationship to the USP Agent
  - Depending on which lower layer protocols are used and whether there are intervening brokers or lower-layer protocol proxies.
- A USP Controller can be in any sort of computing device in a data center, a back office, a smart phone, a laptop, etc.

Lower-layer protocols

- Message Transfer Protocols (MTPs): CoAP, Websockets, STOMP
- Extensible to other MTPs
- Transport Protocols: UDP, TCP
- IP: IPv4, IPv6

The USP Protocol Stack



✓Create
✓Read
✓Update

✓Delete
✓Operate
✓Notify

USP Message
(Protobuf encoding with schema usp-msg.proto)

USP Record
(Protobuf encoding with schema usp-record.proto)

Optional Session Context with TLS

Message Transfer Protocol (MTP)
(CoAP, STOMP, WebSocket)

TLS or DTLS
(depending on MTP)

TCP or UDP
(depending on MTP)

What the protocol stack means for USP

USP Messages and Records

- CRUDON commands: USP includes a set of RESTful messages (Add, Set, Delete, Get, GetInstances, GetSupportedDataModel, and GetSupportedProtocol) plus the Operate and Notify messages, which allow for asynchronous actions and events.
- No more SOAP; no more Remote Procedure Calls
- Operations (firmware update, reboot, file upload, etc.) are now a part of the data model
- Data model information can be addressed by unique key, with wildcards, or with search expressions.
- Failures can be isolated to individual objects and parameters, and relative paths reduce message size significantly.

Role-based access control

- USP defines the trust mechanisms for USP Controllers being associated with USP Agents and role-based access control (access control mechanism defined around roles and privileges) on a per-data-model-element (parameter or object) and per-action (read, write, execute) level that can be managed via the USP data model.

What the protocol stack means for USP

Protocol buffer (protobuf) encoding

- Decreases message size (see https://developers.google.com/protocol-buffers/ for more info)

Optional Session Context with TLS can provide end-to-end security

- The USP Message (inside the USP Record and *above* CoAP / STOMP / Websockets) can be encrypted using TLS 1.2 (soon TLS 1.3) so it is not broken by intermediate proxies or brokers

Flexible, use-case driven transport bindings (MTPs)

- USP's design separates messages and message transport
- USP specification describes how to convey Records over Websockets (long-lived, persistent TCP sessions), STOMP (pub-sub broker sessions) or CoAP (UDP)
- Long-lived sessions mean there is no need to establish a sessions every time a message needs to be sent
- Other mechanisms may be defined in the future, if dictated by a use case

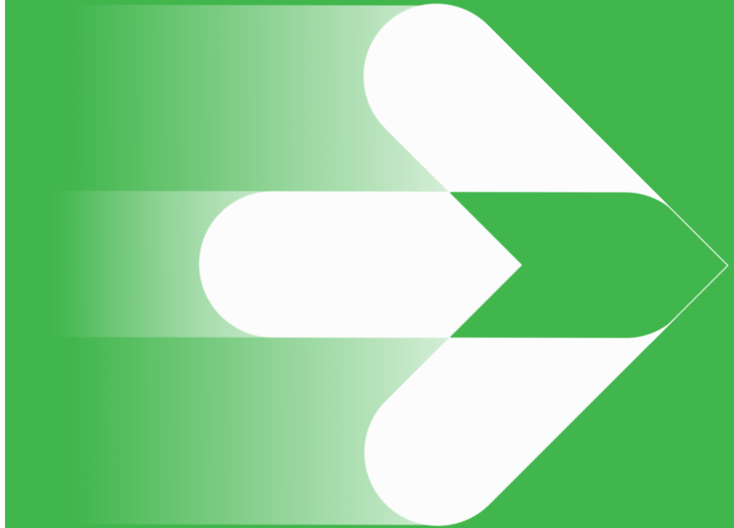
TLS (or DTLS) used to encrypt MTP

- Standard encryption mechanisms supported. If there is a direct link between USP Agents and Controllers, end-to-end security inside the USP Record may not be needed.

Moving it forward, interop and compliance

The **Broadband Forum** schedules regular **plugfests** around USP that allow multiple developers from different companies to work together to harden their implementations.

The **Broadband Forum** is developing both a **certification test plan** including conformance, interoperability, and functional testing, as well as a certification program. Look for more details in the first half of 2019.



Implementation Resources

TR-369 *User Services Platform (USP)* specification at https://usp.technology

The specification for architecture, discovery, end-to-end message encoding, transport, and types, plus security and access control are defined in Broadband Forum TR-369 *User Services Platform (USP)*.

Device:2 data model definitions for USP

The data model for describing the service elements exposed by USP Agents are defined in the Device:2 Root Data Model (published as TR-181 Issue 2). The models for CWMP and USP pull from the same common core with some minor changes for protocol-specific management objects. The models can be found at https://usp-data-models.broadband-forum.org.

Implementation Resources

Protocol buffers schema definitions

USP records and the USP messages they contain are standardized in two “proto” files:
“usp-record.proto” and “usp-message.proto”.
They are linked to on the page at
https://usp.technology/specification/encoding/
or directly at
https://github.com/BroadbandForum/usp/tree/master/specification

For more information on protocol buffers, see
https://developers.google.com/protocol-buffers/.

Thank you

Learn more about the Broadband Forum at:
http://www.broadband-forum.org/

broadband
forum