# IoT Onboarding with DPP





Dan Harkins

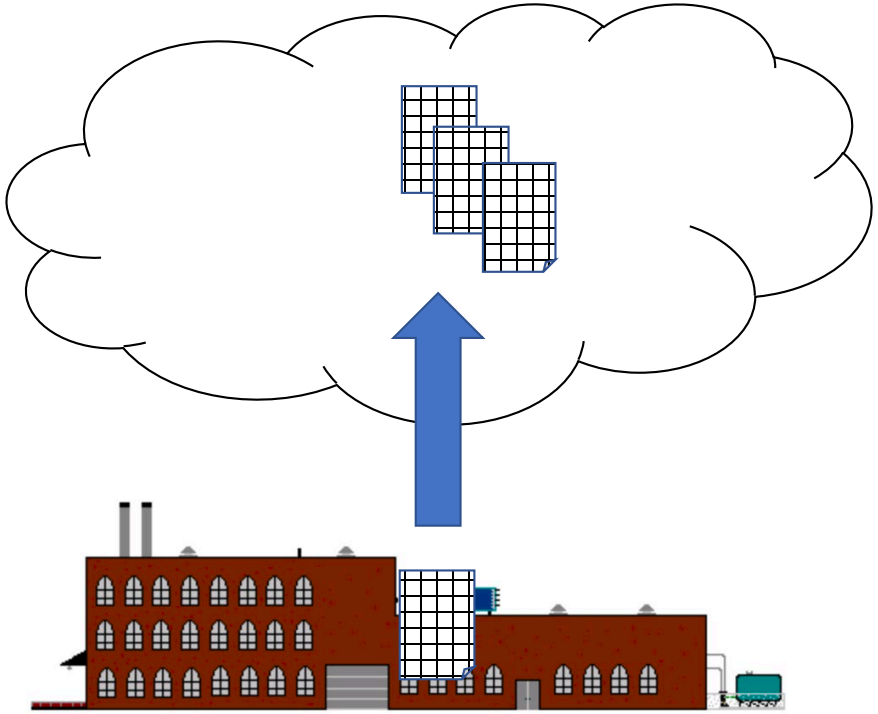
aruba
a Hewlett Packard
Enterprise company

aruba
a Hewlett Packard Enterprise company

# Device Provisioning Protocol

- Robust and secure on-boarding per NIST CSWP on Network-layer onboarding and Lifecycle Management
- Phases of DPP map closely with description of process in NIST CSWP
  - Bootstrapping– establishment of trust in a thing's public key
    - DPP URI contains base64-encoded public key of thing
    - Cloud-based, QR code based, NFC-based, also a PAKE can be used to parlay a simple passcode into a trusted public keys
  - Authentication– strong authentication of device by network, weaker authentication of network by device
  - Provisioning– configuring network credentials in device
  - Network Access– secure connection to network to enable application-layer onboarding
- Uses 802.11 action frames (pre-association, no SSID, no soft-AP)

# Pre-Onboarding with DPP




DPP:C:81/6;K:MDkwEwYHKoZIzj0CAQYIKoZIzj0DAQc….
MUD URL:https://thingbot.example.com/mud

aruba
a Hewlett Packard Enterprise company

# Onboarding for Enterprise

Purchase order transfers ownership of DPP info from vendor cloud

Enterprise network onboarding equipment is able to acquire DPP URIs for all purchased things

No soft-AP so no rogue APs, no extra SSIDs beaconing, on enterprise network

*DPP Presence Announcement* issued by unprovisioned things

- 802.11 action frame consisting of a hash of “chirp” + bootstrapping key
- Network onboarding equipment is able to identify things by chirps
- Only equipment that possesses a thing’s DPP URI is able to provision thing

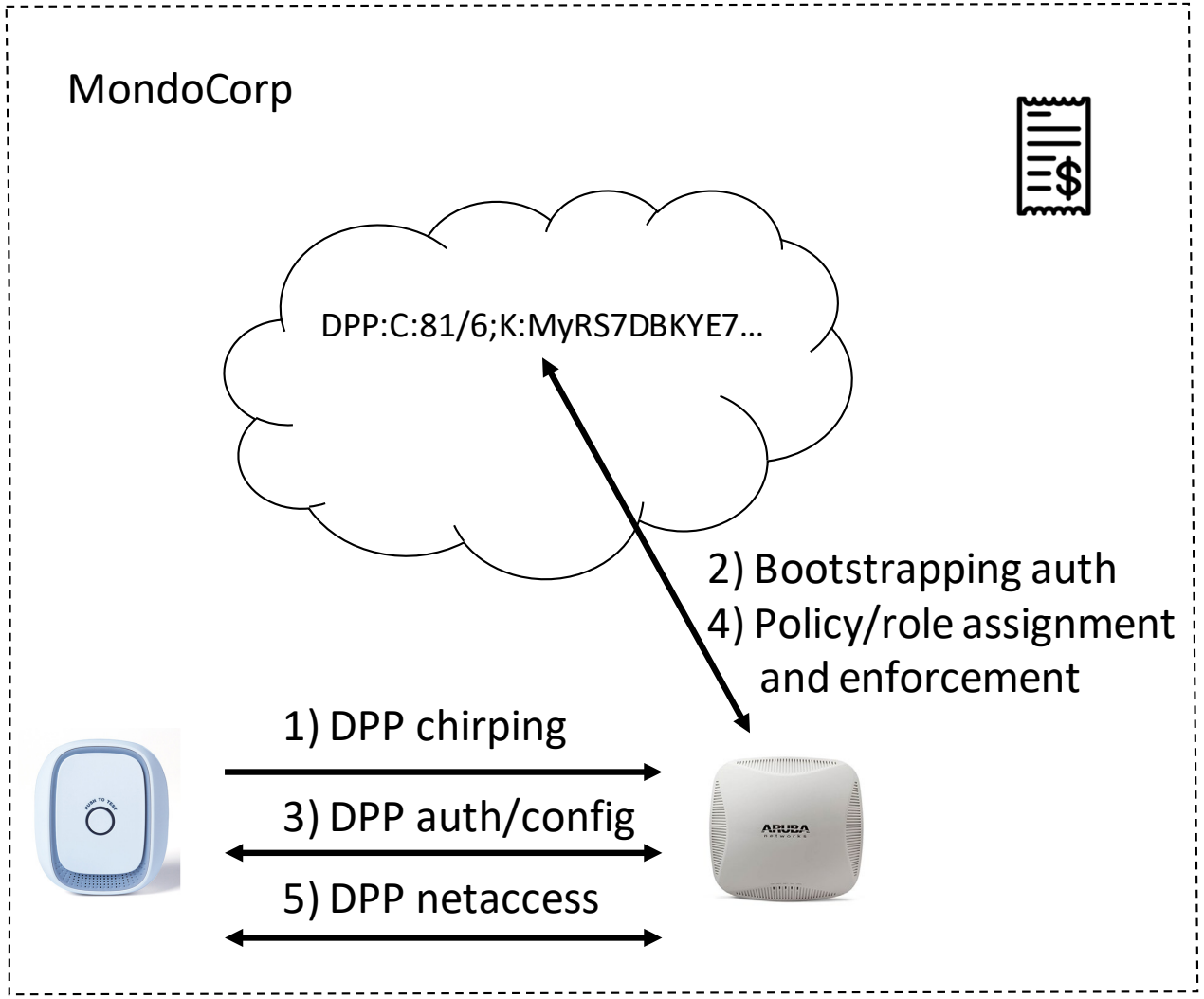
## Trust by Thing

- Manufacturer/vendor trusted to not gratuitously expose bootstrapping key
- The only entity that knows its public bootstrapping key is its legitimate owner

## Trust by Network

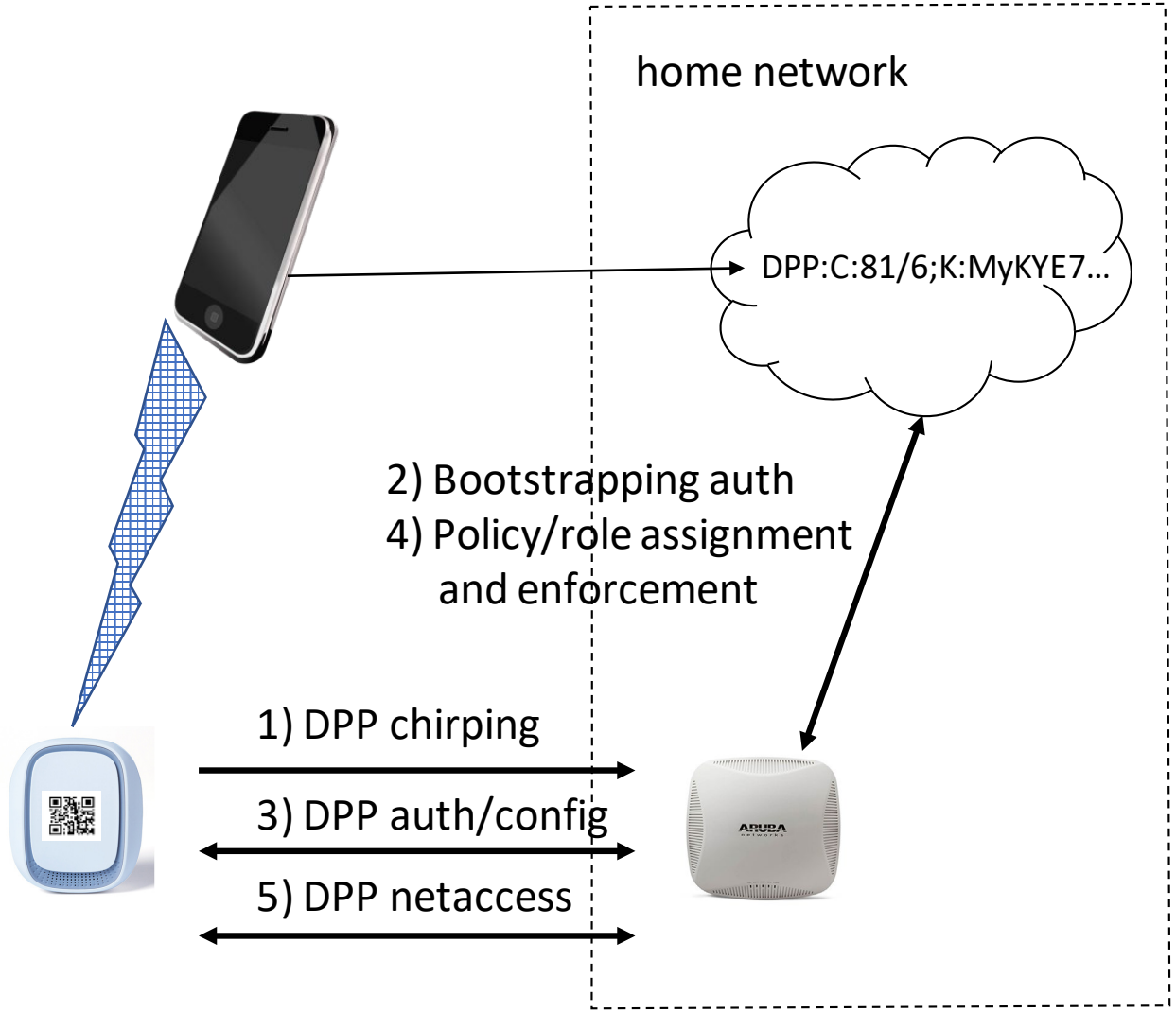
- Vendor/manufacturer is trusted to provide correct DPP URIs for Things
- Thing proves possession of corresponding private key

# DPP Onboarding for Enterprise




- Power-on, device starts chirping
- Identified things are automatically authenticated
- Thing provides MUD URL at provisioning time
- Infrastructure places thing in role and applies policy
- Thing is able to do application-level onboarding
- Secure, robust IoT onboarding at scale

# DPP Onboarding for Home




- Hand-held device scans QR code
- DPP URI transmitted to network infrastructure (e.g. AP or home router or possibly ISP cloud)
  - DNS-SD used to locate endpoint
  - Some authentication still needed
- Policy enforcement possible
- User with no network expertise is able to onboard things securely and robustly

aruba
a Hewlett Packard Enterprise company

# Voucher-less Authorization of Network

## Symmetric trust if thing can do bootstrapping

## Asymmetric trust for headless things*

- Network gets strong cryptographic authentication of thing
- Thing does not get a strong cryptographic authentication of network

## Authorization of network to provision thing depends on thing's trust of bootstrapping method

- The greater the restriction in access to DPP URI the greater the trust
- QR code on backside of thing does not protect against theft of thing, thing knows its being provisioned by physical owner but maybe not legitimate owner
- Cloud-based transfer of ownership can give thing strong statement of authorization due to limited access to its bootstrapping key, thing knows its being provisioned by the legitimate owner
- Manufacturer dictates bootstrapping trust when building thing

* F. Stajano and R. Anderson, The Resurrecting Duckling: Security issues for ad-hoc wireless networks, in: *International Workshop on Security Protocols* (1999).

aruba
a Hewlett Packard Enterprise company

# Network Layer Credentials with DPP

DPP Provisioning provides SSID and credential to thing

Wide support for credentials used in 802.11 today

- Pre-shared key for WPA2-PSK AKM
- Password for WPA3-SAE AKM
- X.509 certificate for WPA3-Enterprise (including WPA3-CNSA) AKM
  - RSA and ECC support
  - RFC 7030-style CSR Attributes request
- Connector– a signed ECC public key for DPP AKM

Network access with DPP Connector

- Client and AP exchange connectors signed by same authority
- Client and AP do Diffie-Hellman using public key from peer's connector
- Resulting secret becomes PMK, then 4-way handshake, etc

aruba
a Hewlett Packard Enterprise company

# DPP Ease-of-Use

DPP workflow is, “plug it in, turn it on…it works”

Misuse resistance: easy to use correctly, difficult to use incorrectly

- QR codes scan or they don’t, once scanned there is nothing else to do
- Manufacturers and vendors have ownership transfer of things worked out

No IoT or networking expertise needed to onboard things

- Industrial deployment (e.g. nuclear power plant) allow for things to be installed by a crew with no IT skills– just mount the device, apply power
- Homeowner just unpacks, scans, plugs device in
- Chirping device will be discovered and provisioned automatically

No rigid onboarding process to follow– bootstrapping can take place before or after device is installed

Onboarding at scale

aruba
a Hewlett Packard Enterprise company

# DPP Bootstrapping for Wired Devices

DPP is an 802.11 protocol

Wired devices on enterprise (802.1X/EAP) have classic catch-22– you must have a certificate to get on the network and you need to get on the network to get a certificate

TLS-pok– “proof of knowledge”– leverages DPP bootstrapping for TLS

- Works with any DPP bootstrapping method– cloud, QR code, NFC, etc.
- Defined in Internet-Draft draft-friel-tls-eap-dpp

TEAP + TLS-pok

- Establish “outer” tunnel with DPP bootstrapping and TLS-pok
- Use “inner” tunnel to enroll in enterprise CA with existing TEAP functionality

Subsequent network access by device uses

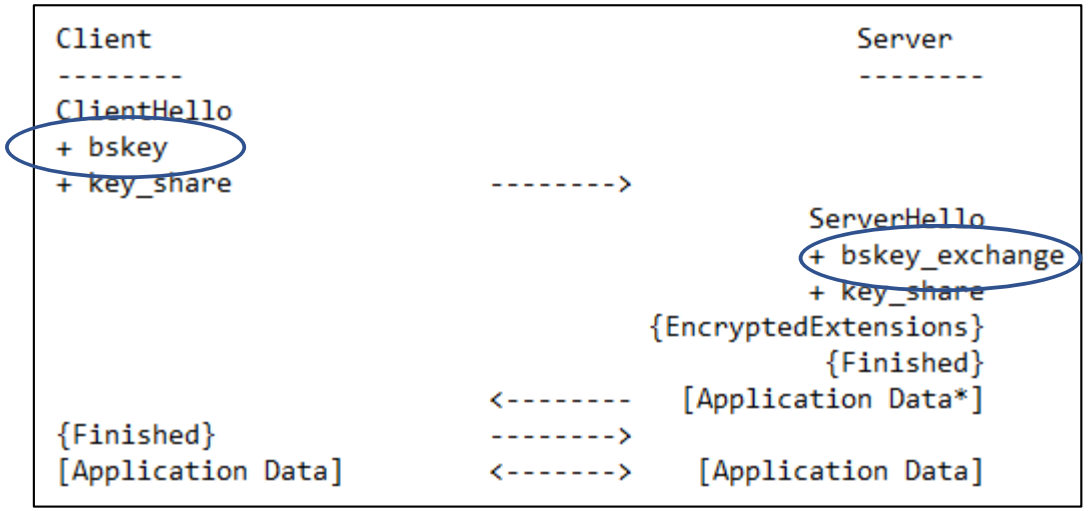
# Handshaking Changes for TLS-pok

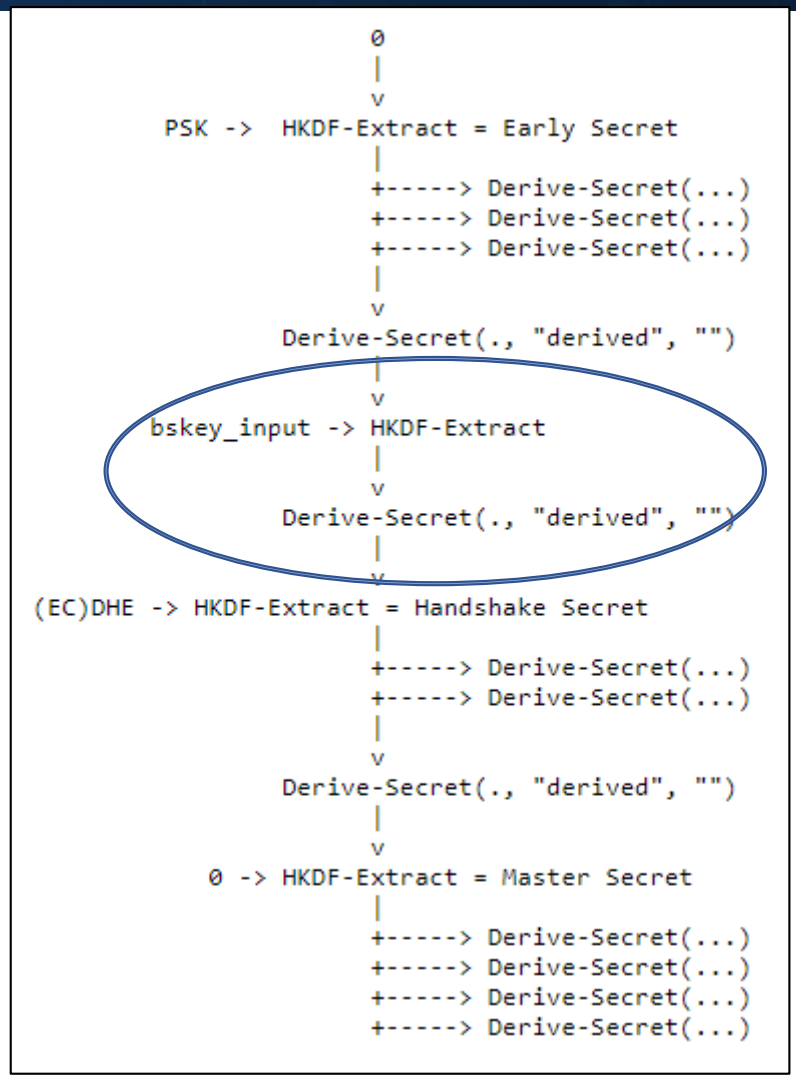
```
struct {
    select (Handshake.msg_type) {
        case client_hello:
            opaque bskey[32];

        case server_hello:
            opaque bskey_exchange<1..2^16-1>;
    };
} BootstrapKey;
```

The BootstrapKey extension is used by the client in its ClientHello message to specify its bootstrapping key identifier. The 'bskey' field of this extension SHALL consist of the base64 encoded SHA256 digest of the DER-encoded ASN.1 subjectPublicKeyInfo representation of the bootstrapping public key.

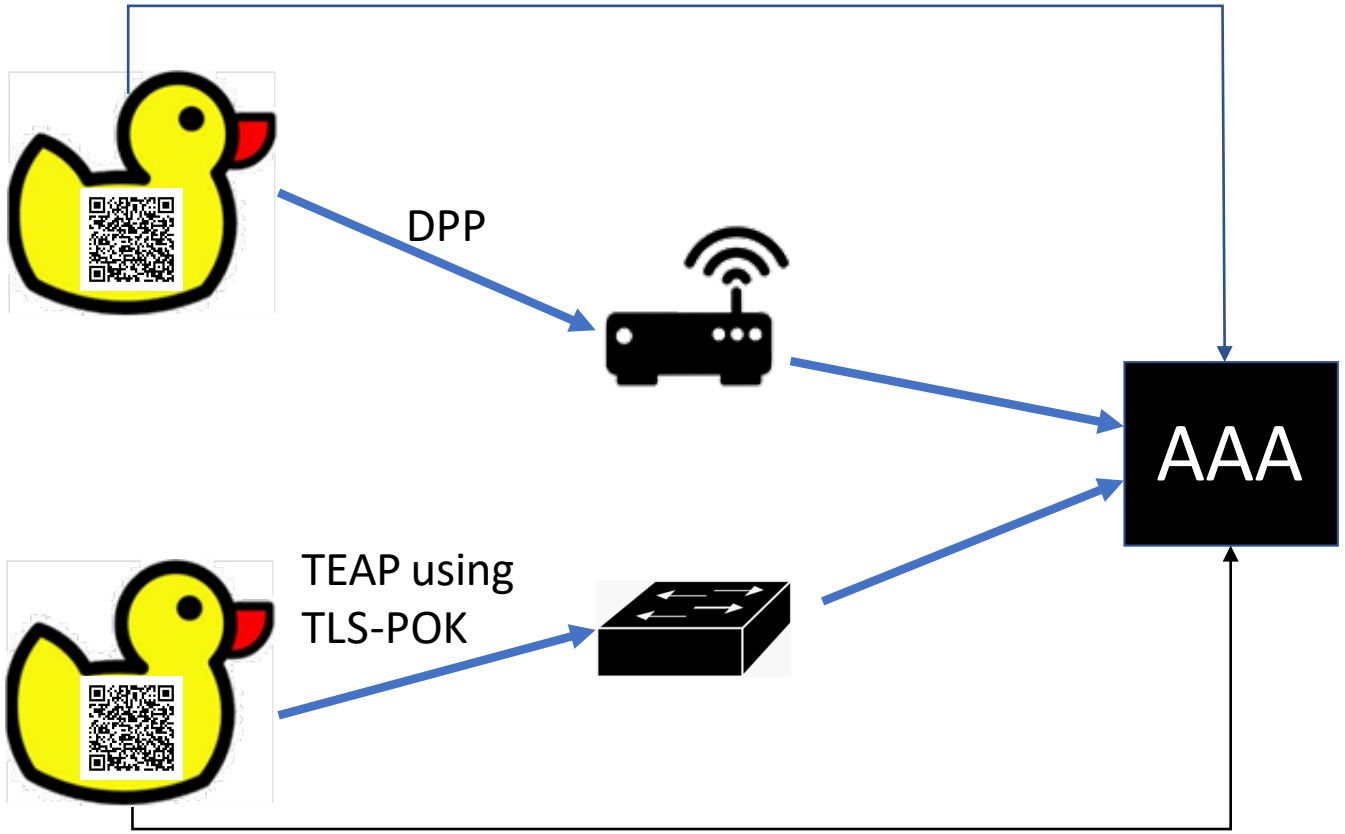
The BoostrapKey extension is used by the server in its ServerHello message to specify its ephemeral ECDH keying information. The 'bskey_exchange' field contains the key exchange information on the curve that the bootstrapping key is on.




"proof of knowledge" ciphersuite for TLS 1.3

# Everyone is Happy!




Import/authenticate/provision

aruba
a Hewlett Packard Enterprise company

# Resources

## DPP Specification

https://www.wi-fi.org/file/device-provisioning-protocol-specification

## TLS-pok Specification

https://datatracker.ietf.org/doc/draft-friel-tls-eap-dpp/

## Reference implementations (free code!)

https://github.com/HewlettPackard/dpp

https://github.com/upros/mint/tree/tls-pok

# *Thank You!*

Dan Harkins

aruba

a Hewlett Packard
Enterprise company