

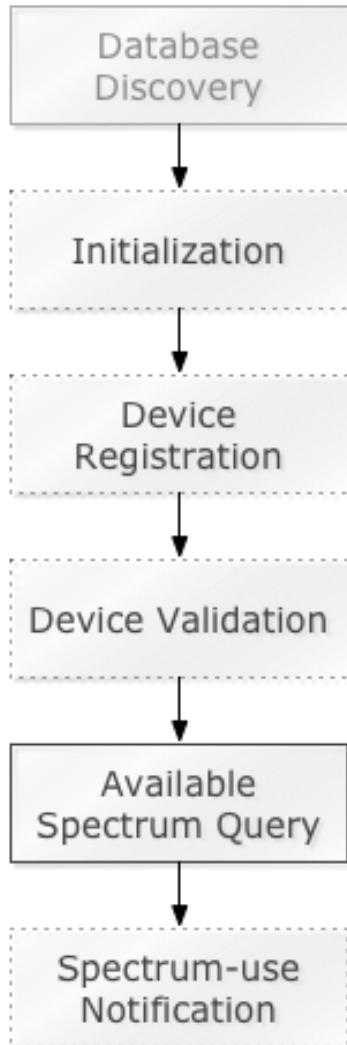
Outline

- What's in the document
- Open Issues
 - Encoding
 - Device Discovery
 - Security

What's in the document

- Clarified whether messages and parameters are:
 - Mandatory
 - Optional
 - Regulator-specific

Protocol Overview



- Database Discovery (TBD)
- Initialization
 - Initial handshake: Exchange capability info
 - Optional for device, DB must support
- Device Registration
 - Regulator-specific (e.g., not all regulators)
- Device Validation
 - Validation of slave devices by master devices
 - Regulator-specific
- Available Spectrum Query
- Spectrum-use Notification
 - Regulator-specific

Initialization

- Purpose:
 - To exchange capability information
 - Allow database implementations or regulatory domains add extra handshake.
- Request parameters
 - Common parameters
 - Device identifier
 - Location
 - Optional regulatory-specific data (e.g., Ofcom technology)
 - Optional database-specific data
- Response parameters
 - Common rule-set parameters (see next slide)
 - Optional regulatory-specific data
 - Optional database-specific data

Common Rule-set Info

- Regulatory domain, e.g., “us”, “uk”, etc
- Thresholds beyond which device must request new spectrum information
 - Maximum polling interval
 - Maximum location change
 - Maximum validity duration
- NOTE: Initialization is optional, because these rules could be statically configured in a device

Encoding Open Issues

- JSON-RPC
- Regulatory-specifics
- Encoding Examples
- Geo Location
- vCard

For Consideration: JSON-RPC

- Specification (<http://www.jsonrpc.org>)

- Request

- ```
{"method": "echo", "params": ["Hello"], "id": 1}
```

- Response

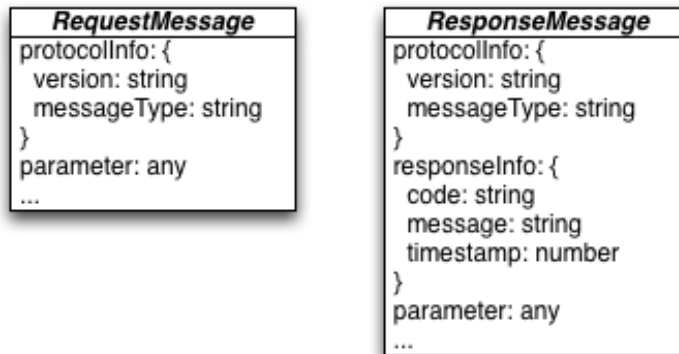
- ```
{"result": "Hi", "errors": null, "id": 1}
```

- Natural fit for PAWS Protocol

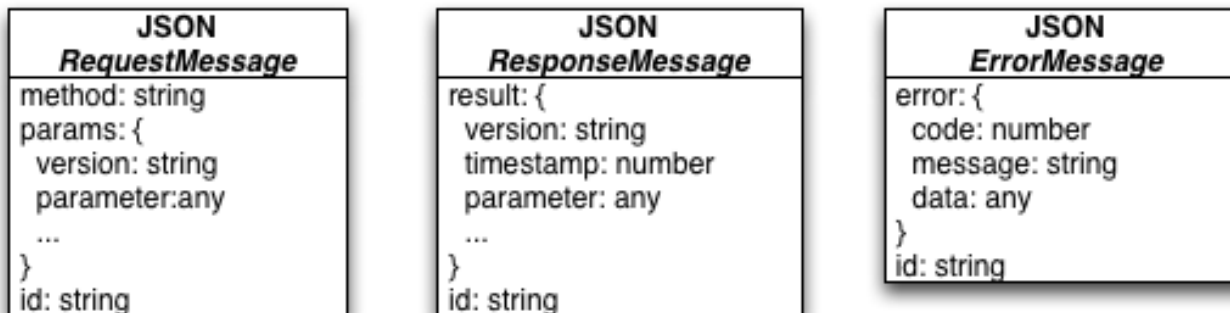
- “params” for request messages
 - “result” for response messages
 - “errors” for error codes and messages

JSON-RPC Comparison

- Currently defined message structure
 - Extra layer: protocolInfo and responseInfo



- JSON-RPC message structure



Indicating Errors

- Current draft does not define error codes
- Proposal: Keep PAWS and HTTP layers separate
 - Still 200 OK at the HTTP layer
 - Response codes
 - OK
 - VERSION
 - UNSUPPORTED
 - UNIMPLEMENTED
 - UNAUTHORIZED
 - REQUIRED
 - INVALID_VALUE
 - OUTSIDE_COVERAGE
 - NOT_REGISTERED

Encoding: Regulatory Specifics

- Define regulatory specifics in appendix
 - Payload of regulatory-domain parameter: Normative or informative?
- Example encoding for “device identifier” with extensibility for regulator-specific info:

```
{
  "serialNumber": "0234D87654554D87C",
  "us": {
    "fccId": "OPS13",
    "deviceType": "FIXED",
  },
  "uk": {
    ...
  }
}
```

Encoding: JSON Schema

- INIT_MSG schema

```
{
  "name": "INIT_REQ",
  "description": "Initialization Request message.",
  "type": "object",
  "properties": {
    "protocolInfo": {
      "type": "ProtocolInfo",
      "required": true
    },
    "deviceId": {
      "type": "DeviceIdentifier",
      "required": true
    },
    "location": {
      "type": "GeoLocation",
      "required": true
    }
  }
}
```

Encoding: JSON Schema

- ProtocolInfo Schema

```
{  
  "name": "ProtocolInfo",  
  "description": "Required information for all PAWS messages.",  
  "type": "object",  
  "properties": {  
    "version": {  
      "description": "Version of the PAWS protocol",  
      "type": "string",  
      "required": true  
    },  
    "messageType": {  
      "description": "Name of the containing message (e.g., INIT_REQ)",  
      "type": "string",  
      "required": true  
    },  
  },  
}
```

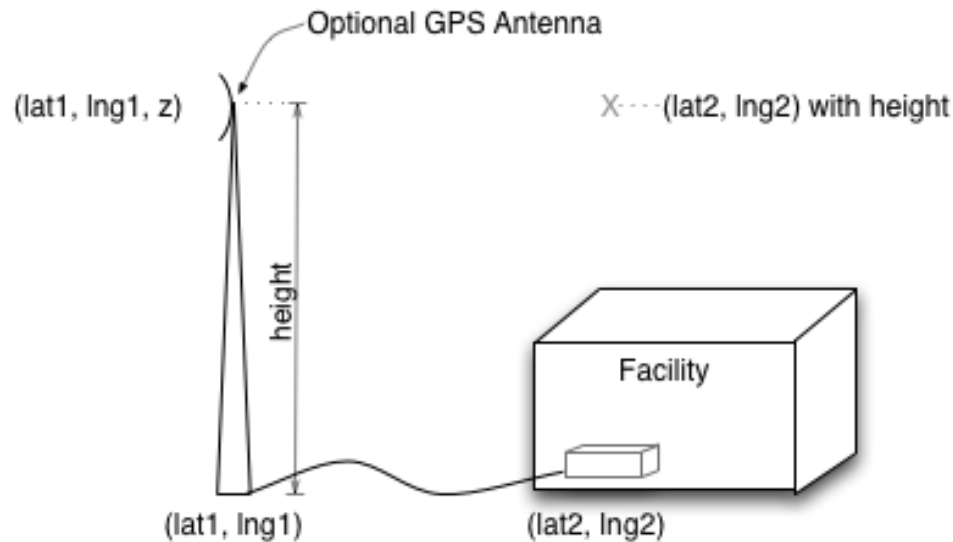
Encoding: JSON Example

- INIT_MSG

```
{
  "protocolInfo": {
    "version": "1.0",
    "messageType": "INIT_MSG"
  },
  "deviceId": {
    "serialNumber": "0234D87654554D87C",
    ...
  },
  "location": {
    "latitude": 37.9938,
    "longitude": -120.09384,
    ...
  }
}
```

Encoding: Geolocation

- Location of Device vs Antenna



- Intent: Compute available spectrum at $(lat1, lng1, z)$
 - Z Relative to ground: Measure by installer
 - Z Relative to sea level: Automated measurement, e.g., via GPS

Encoding: Geolocation

- RFC6225 GeoLoc: latitude, longitude, altitude with uncertainties
 - Latitude, uncertainty
 - Longitude, uncertainty
 - Altitude, uncertainty
 - datum
- Should antenna height be specified separately within antenna characteristics? or use a single “location” object?

Batch Requests

- The draft allows batch requests (optional) for a sequence of locations
 - Use case: Mobile device asking for spectrum along anticipated path
 - The Database MAY return responses in any order
 - The Database MAY limit the number of locations processed
 - NOTE: Does not satisfy requirement D8
 - Requires support for a geographic area

How to Support requirement D8?

- Option 1: Extend RFC6225
 - Add radius to support circle
 - List of points (lat, long) to support polygon
- Option 2: JSON encode RFC5491
 - Only need circle and polygon shapes

Encoding: vCard

- draft-bhat-vcarddav-json-00 provides JSON encoding of vCard
- Only contact fields will be used, e.g.,

```
{
  "version": "4.0",
  "fn": "John Smith",
  "adr": {
    "street": "100 Main Street",
    "locality": "Summersville",
    "region": "CA",
    "code": "90034",
    "country": "USA"
  },
  "tel": {
    "uri": "tel:+1-213-555-2344"
  }
  "email": {
    "text": "j.smith@email.com"
  }
}
```

Database Discovery

- Should it be a different document?

Security: Observations

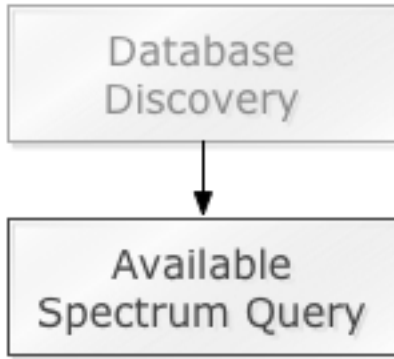
- A malicious device can use spectrum without ever contacting a database
- No current White Space rules (FCC, Ofcom) require client authentication

Security: Client Authentication

- Challenge: Relies on physical security of millions of devices:
 - With access to the physical device, secrets can be extracted, allowing impersonation
- Given that a malicious device can operate without even contacting the Database, what is the role of client authentication?

Backup Slides

Protocol Overview: Basic



- Database Discovery (TBD)
- Available Spectrum Query
 - Device provides its identity and location
 - Database responds with schedule of available spectrum, as determine by regulator rules

Example JSON: RFC6225 GeoLoc

```
{  
  "latitude": 37.231923,  
  "latUnc": 25,  
  "longitude": -120.39485,  
  "longUnc": 25,  
  "altitude": 52,  
  "altUnc": 10,  
  "atype": 1,  
  "datum": "WGS84"  
}
```