DATA PLANE

API

Uplink data from a Device via a Network Server

```
POST /api/uplink/:applicationId/:networkId
```

```
{
   applicationID: 'string',
   applicationName: 'string',
   deviceName: 'string',
   devEUI: 'string, base64',
   rxInfo:[
   {
       gatewayID: 'string, base64',
       name: 'string',
       time: 'string, time-ISO',
       rssi: 'number',
       loRaSNR: 'number',
       location:{
           latitude: 'number',
           longitude: 'number',
           altitude: 'number'
        }
   }],
   txInfo:{
       frequency: 'number',
       dr: 'number'
   },
   adr: 'boolean',
   fCnt: 'number',
   fPort: 'number',
   data:'string, base64 e.g. eyJXRCI6ICJOVyIsICJIIC...',
}
```

Downlink to All Devices

```
POST /api/downlink/:applicationId
```

```
{
  "deviceQueueItem": {
    "confirmed": true,
    "data": "string",
    "fCnt": 0,
    "fPort": 0,
    "jsonObject": "string"
  }
}
```

Downlink to a Single Device

```
POST /api/downlink/:applicationId/:devEUI
```

```
{
  "deviceQueueItem": {
    "confirmed": true,
    "data": "string",
    "devEUI": "string",
    "fCnt": 0,
    "fPort": 0,
    "jsonObject": "string"
  }
}
```

Application Server Payload

Note Application Server API is application dependent

```
  {
    applicationID: 'string',
    applicationName: 'string',
    deviceName: 'string',
    devEUI: 'string, base64',
```

```
        rxInfo:[
        {
            gatewayID: 'string, base64',
            name: 'string',
            time: 'string, time-ISO',
            rssi: 'number',
            loRaSNR: 'number',
            location:{
                latitude: 'number',
                longitude: 'number',
                altitude: 'number'
             }
        }],
        txInfo:{
            frequency: 'number',
            dr: 'number'
        },
        adr: 'boolean',
        fCnt: 'number',
        fPort: 'number',
        data:'string, base64 e.g.
    eyJXRCI6ICJOVyIsICJIIjogIjAiLCAidGltZSI6IC...',
      }
```

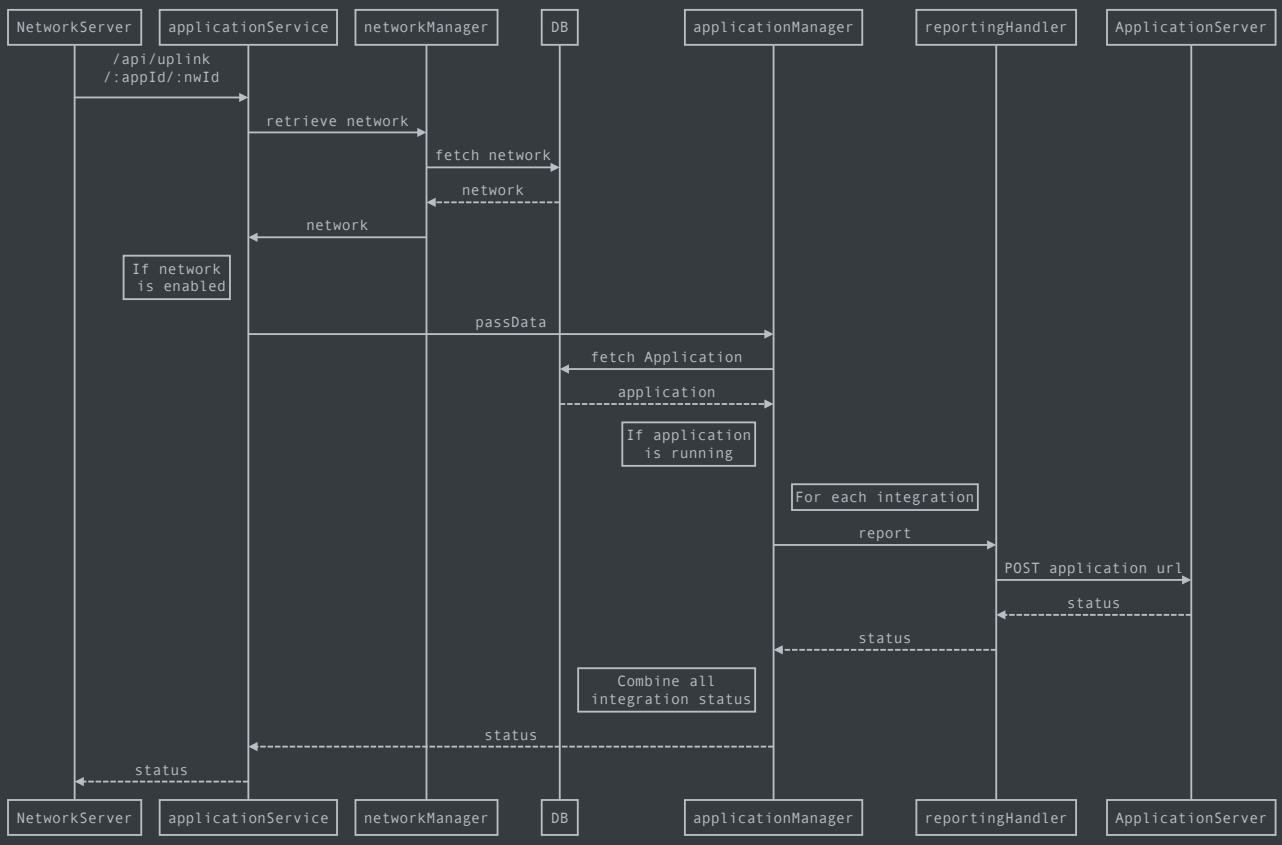
The purpose of the network is to see if it is enabled or not. If not, the data is dropped The purpose of the application is 1 To find out if the application is running (if no drop data) 2 To find out the integration information

Requirements

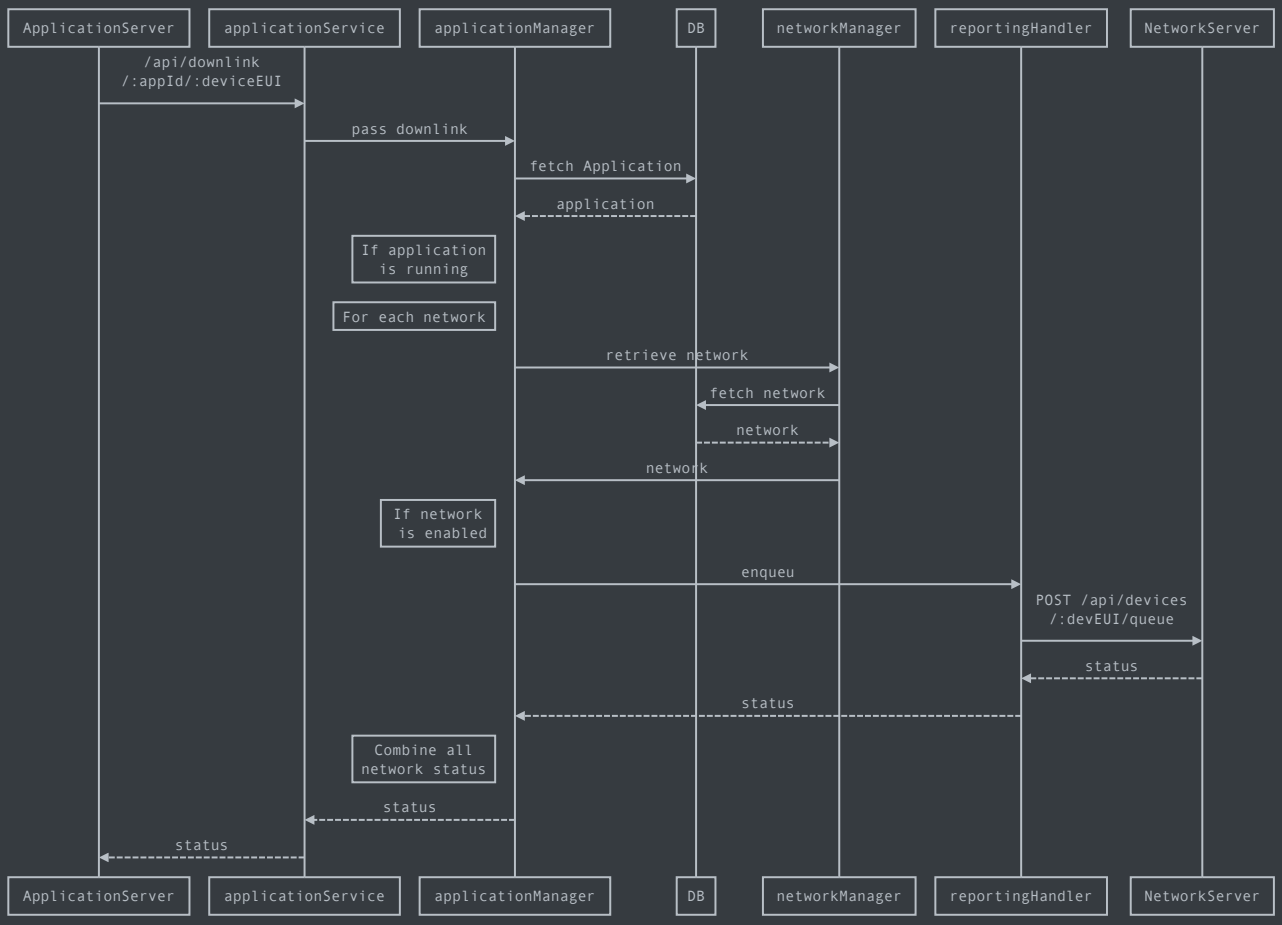
- Should support Downlink as well as uplink
- Should allow a single application to support multiple integrations
- Should support payload analysis for trouble shooting in UI

Proposed Flow

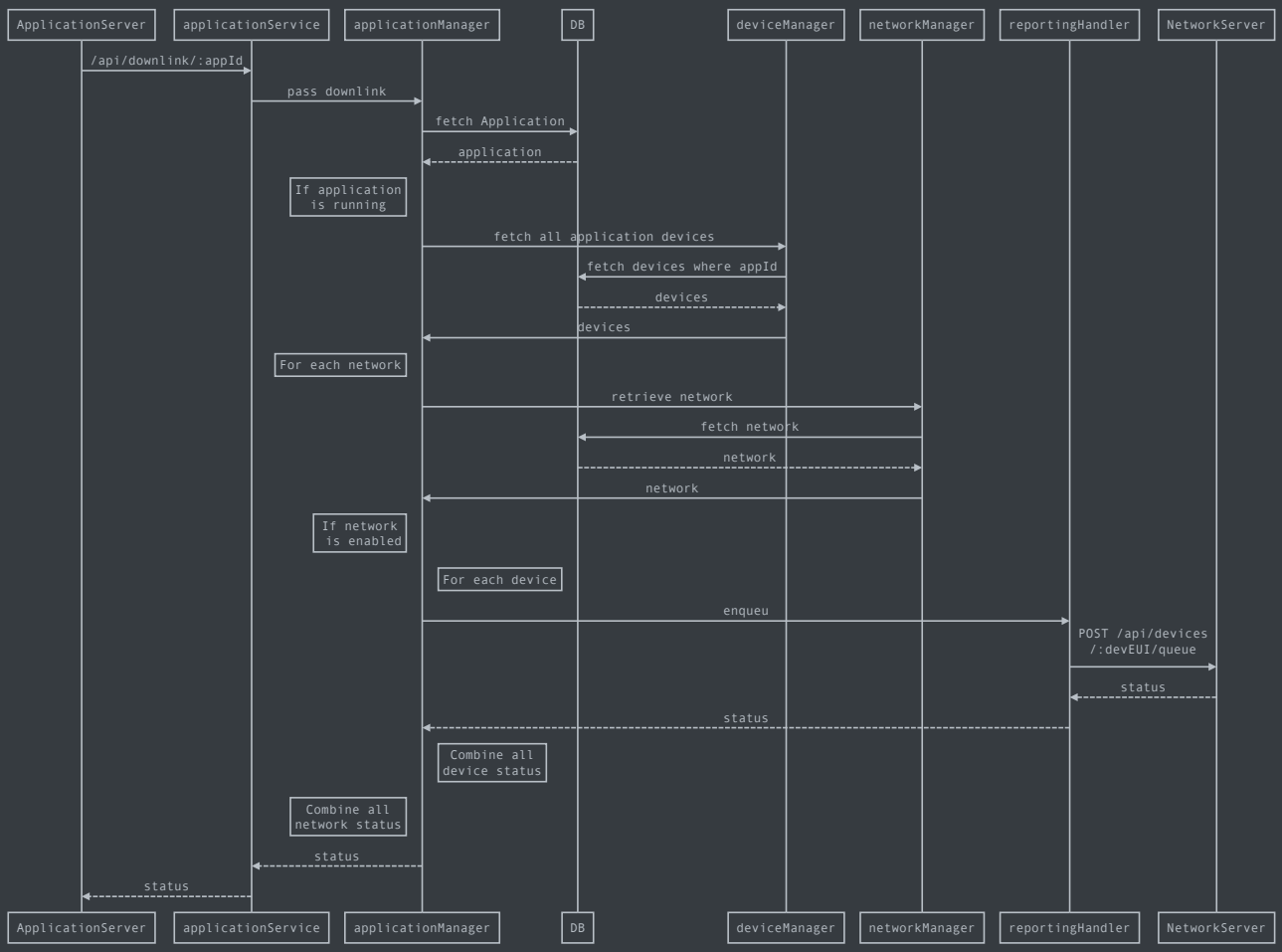
Uplink



Downlink to Single Device



Downlink to All Devices



Current Implementation

API

Uplink data from a Device via a Network Server

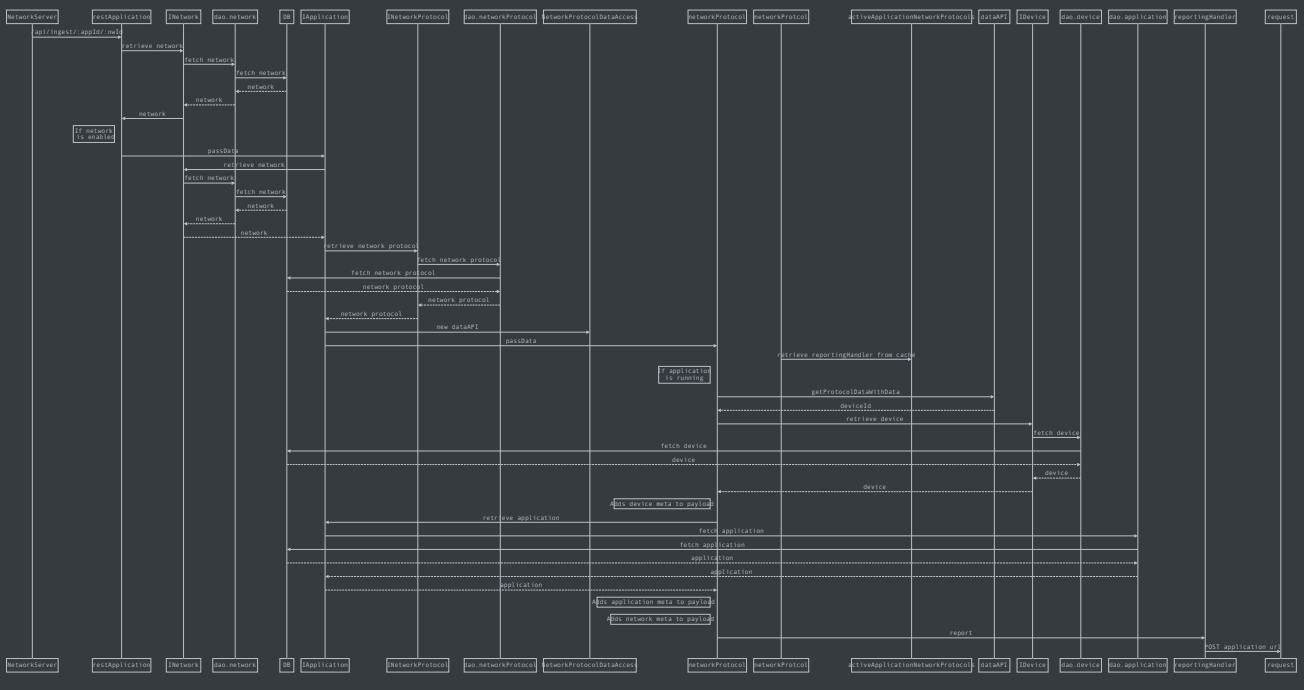
```
POST /api/ingest/:applicationId/:networkId
```

```
{
    applicationID: 'string',
    applicationName: 'string',
    deviceName: 'string',
    devEUI: 'string, base64',
    rxInfo:[
    {
        gatewayID: 'string, base64',
```

```
            name: 'string',
            time: 'string, time-ISO',
            rssi: 'number',
            loRaSNR: 'number',
            location:{
                latitude: 'number',
                longitude: 'number',
                altitude: 'number'
             }
        }],
        txInfo:{
            frequency: 'number',
            dr: 'number'
        },
        adr: 'boolean',
        fCnt: 'number',
        fPort: 'number',
        data:'string, base64 e.g.
    eyJXRCI6ICJOVyIsICJIIjogIjAiLCAidGltZSI6IC...',
      }
```

Current Flow

Note that Downlinks are not supported in the current codebase



Application Server Payload

Note Application Server API is application dependent

```
    {
      applicationID: 'string',
      applicationName: 'string',
      deviceName: 'string',
      devEUI: 'string, base64',
      rxInfo:[
      {
          gatewayID: 'string, base64',
          name: 'string',
          time: 'string, time-ISO',
          rssi: 'number',
          loRaSNR: 'number',
          location:{
              latitude: 'number',
              longitude: 'number',
              altitude: 'number'
           }
      }],
      txInfo:{
          frequency: 'number',
          dr: 'number'
      },
      adr: 'boolean',
      fCnt: 'number',
      fPort: 'number',
      data:'string, base64 e.g.
  eyJXRCI6ICJOVyIsICJIIjogIjAiLCAidGltZSI6IC...',
      deviceInfo: {
          name: 'string',
          description: 'string',
          model: 'string'
      },
      applicationInfo: {
          name: 'string'
      },
```

```
    networkInfo: {
        name: 'string'
    }
}
```