



**Activity**  
Connecting with intelligence

## DX Location API for integrating

---

2019-04-17

- Abeeway Trackers
- ThingPark Location Solver
- 3<sup>rd</sup> Party Application Server
- 3<sup>rd</sup> Party Network server



# Agenda

- ThingPark X Location Engine message flows
- Hands-on exercise

Actility

# Agenda

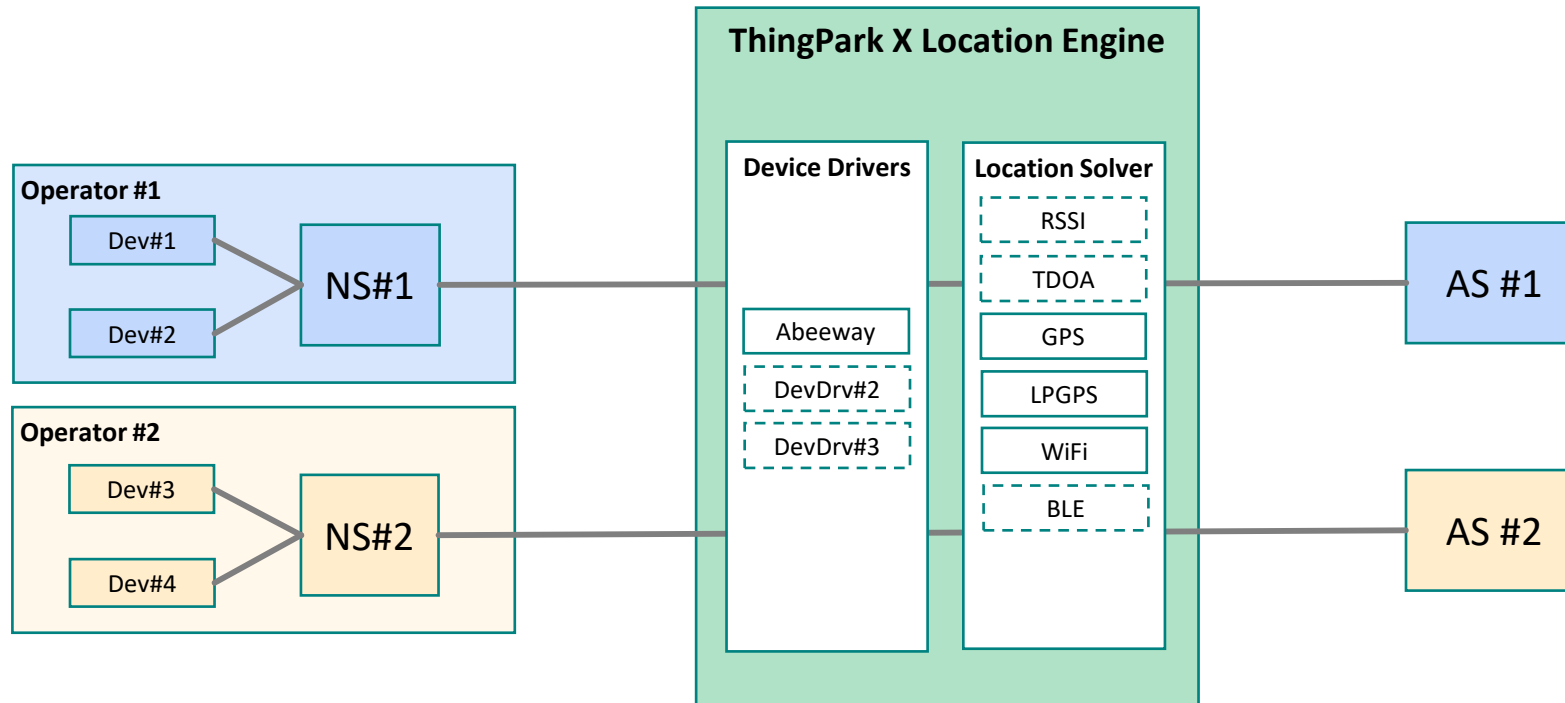
- ThingPark X Location Engine message flows
- Hands-on exercise

Activity

# ThingPark X Location Engine

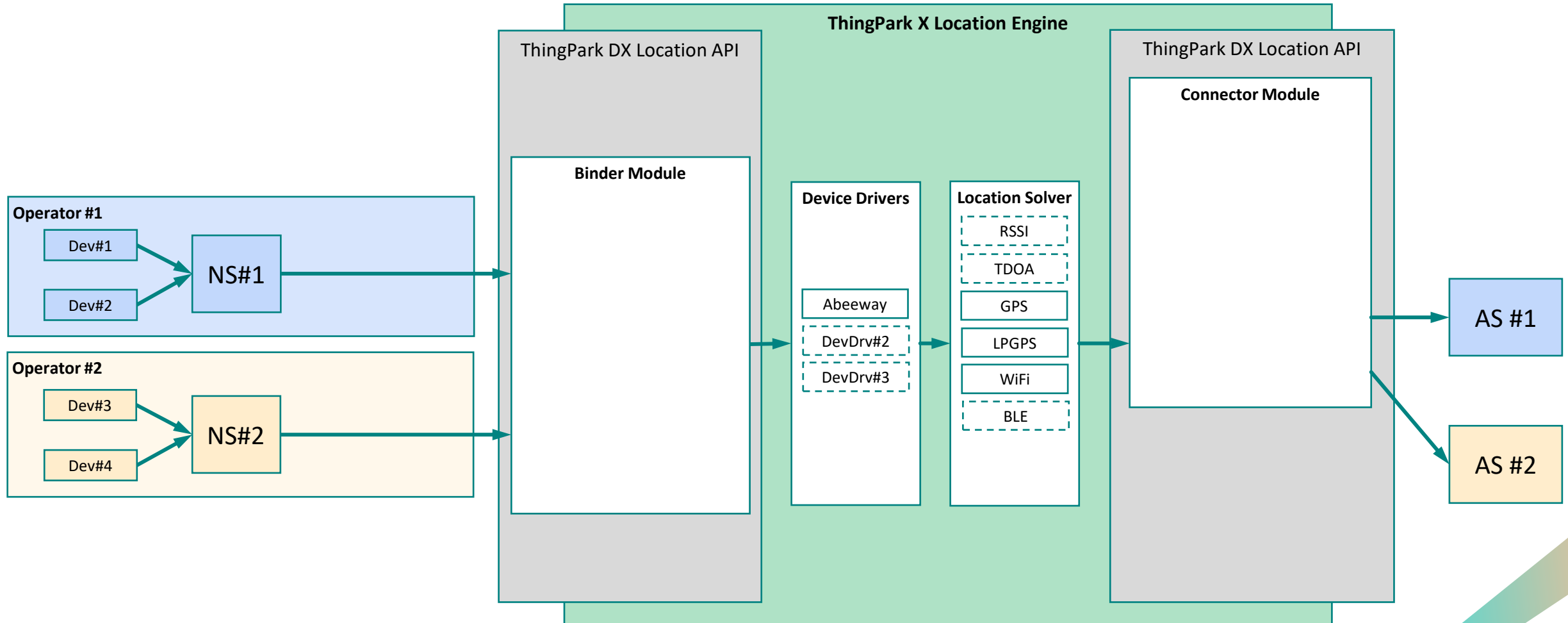
## Multi-technology Location Solver

TDOA/RSSI Solver on TPX LE  
will be supported in **Q3 2019**



# ThingPark X Location Engine and DX Location API

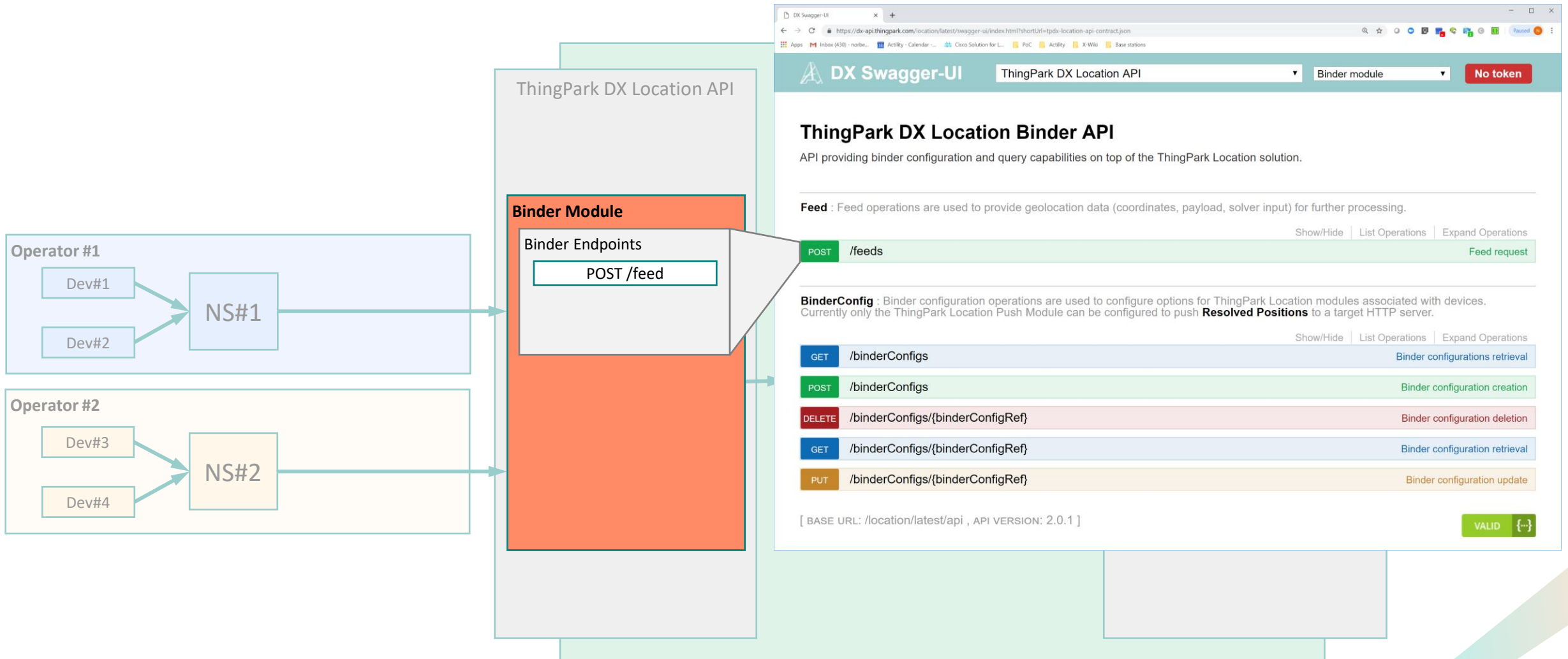
## Uplink Routing: High level view



# ThingPark X Location Engine and DX Location API

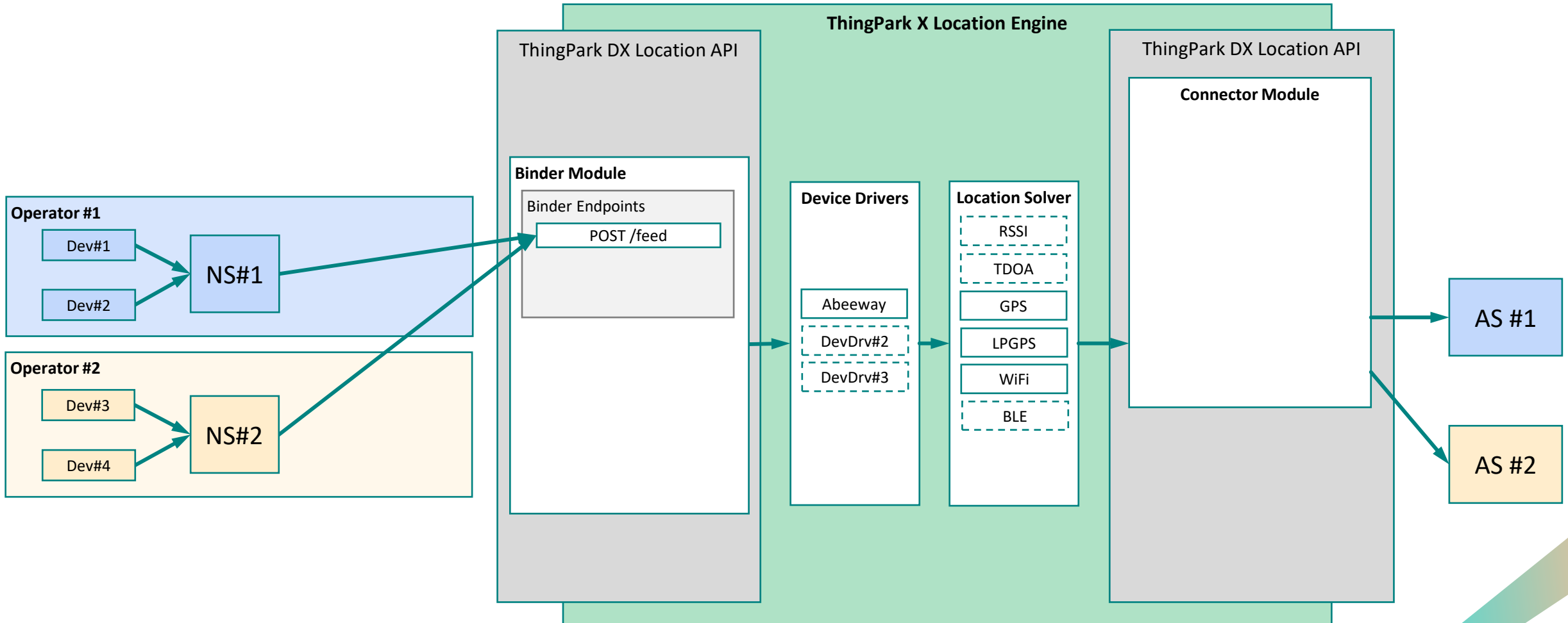
## Uplink Routing: High level view

<https://dx-api.thingpark.com/location/latest/swagger-ui/index.html?shortUrl=tpdx-location-api-contract.json>



# ThingPark X Location Engine and DX Location API

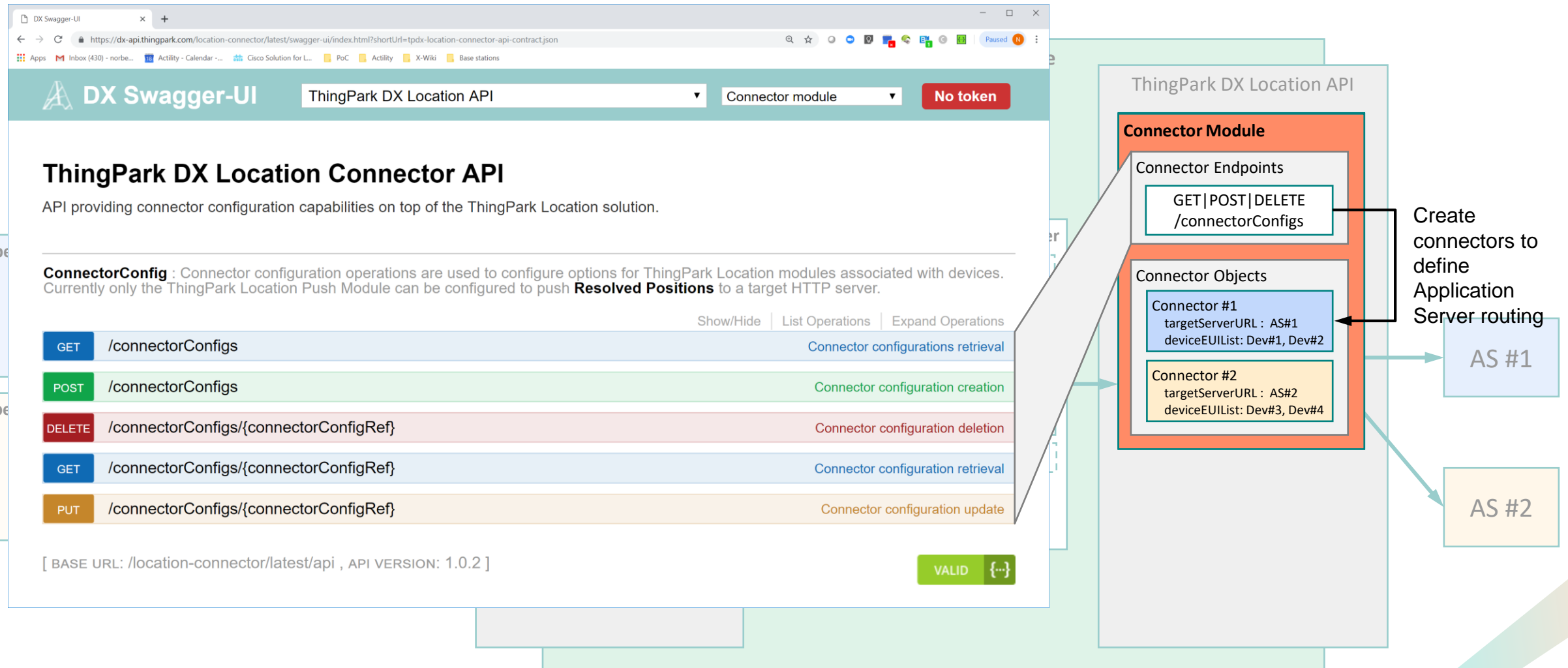
## Uplink Routing: Connector Module



# ThingPark X Location Engine and DX Location API

## Uplink Routing: Connector Module

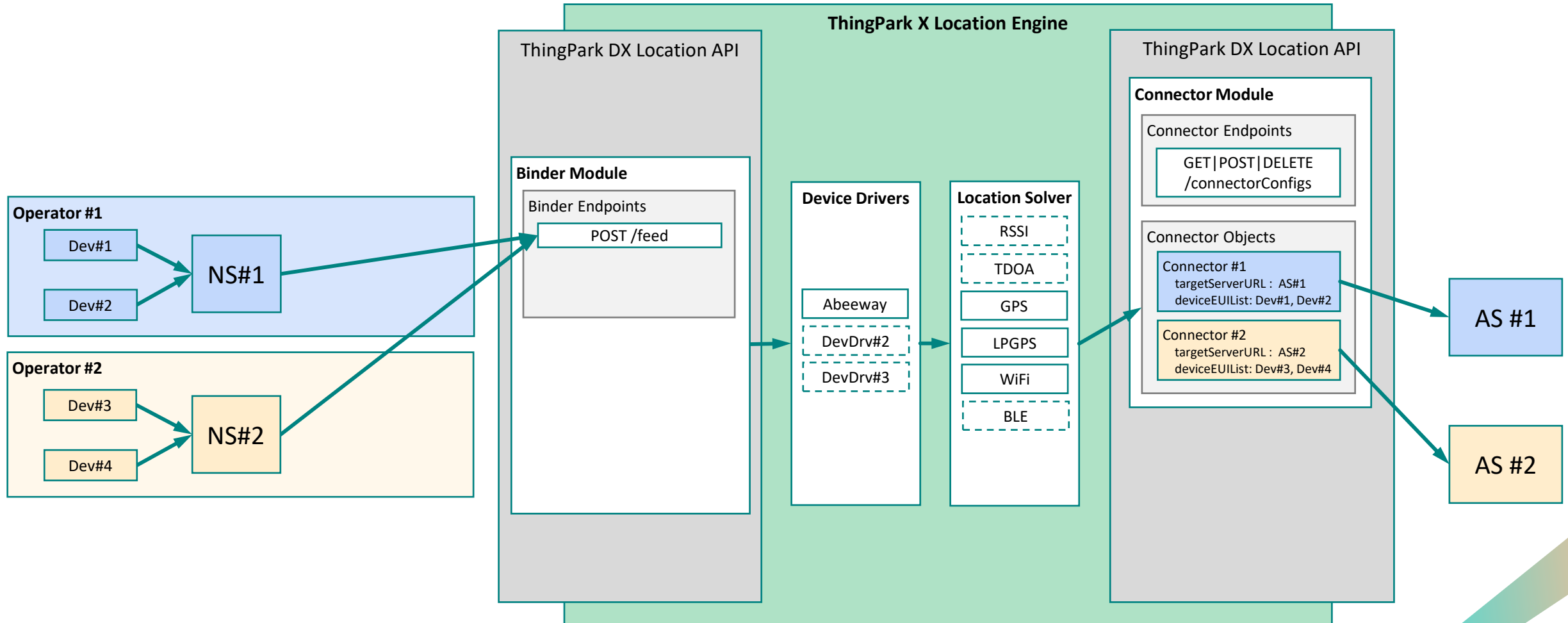
<https://dx-api.thingpark.com/location-connector/latest/swagger-ui/index.html?shortUrl=tpdx-location-connector-api-contract.json>





# ThingPark X Location Engine and DX Location API

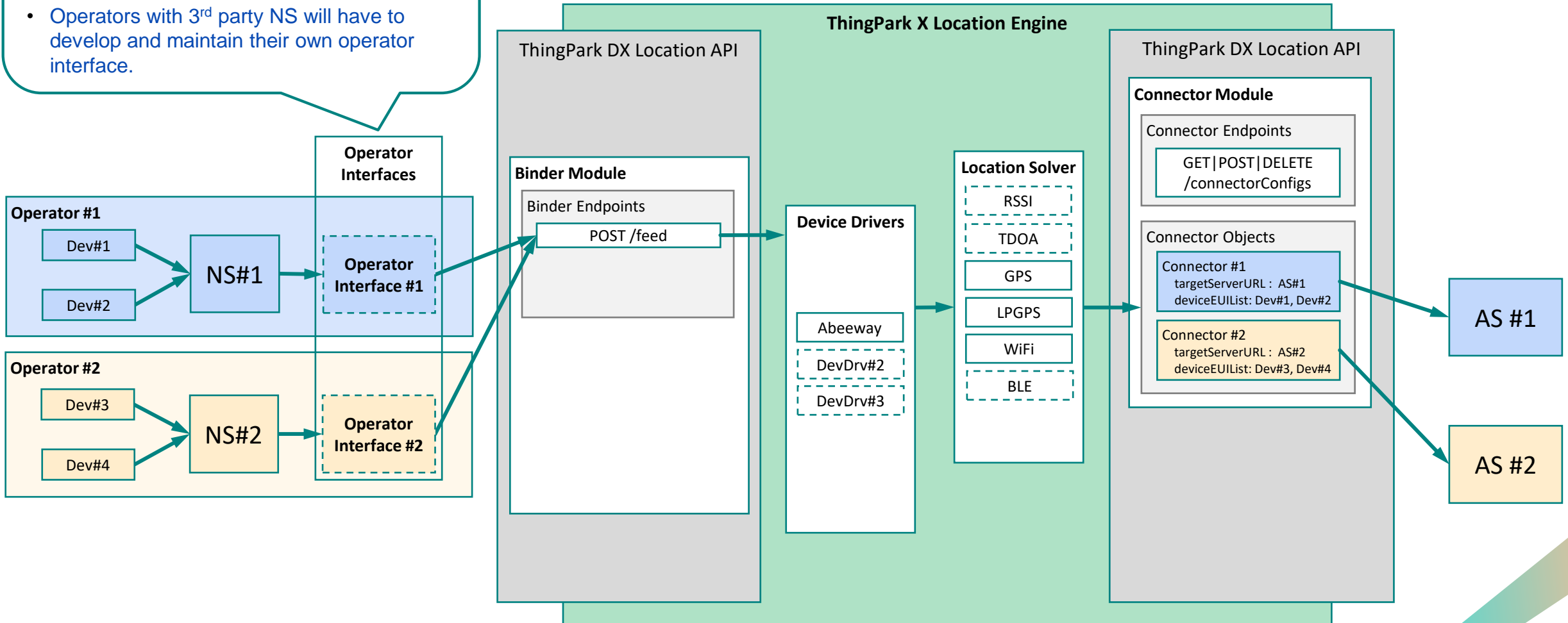
## Uplink Routing



# ThingPark X Location Engine and DX Location API

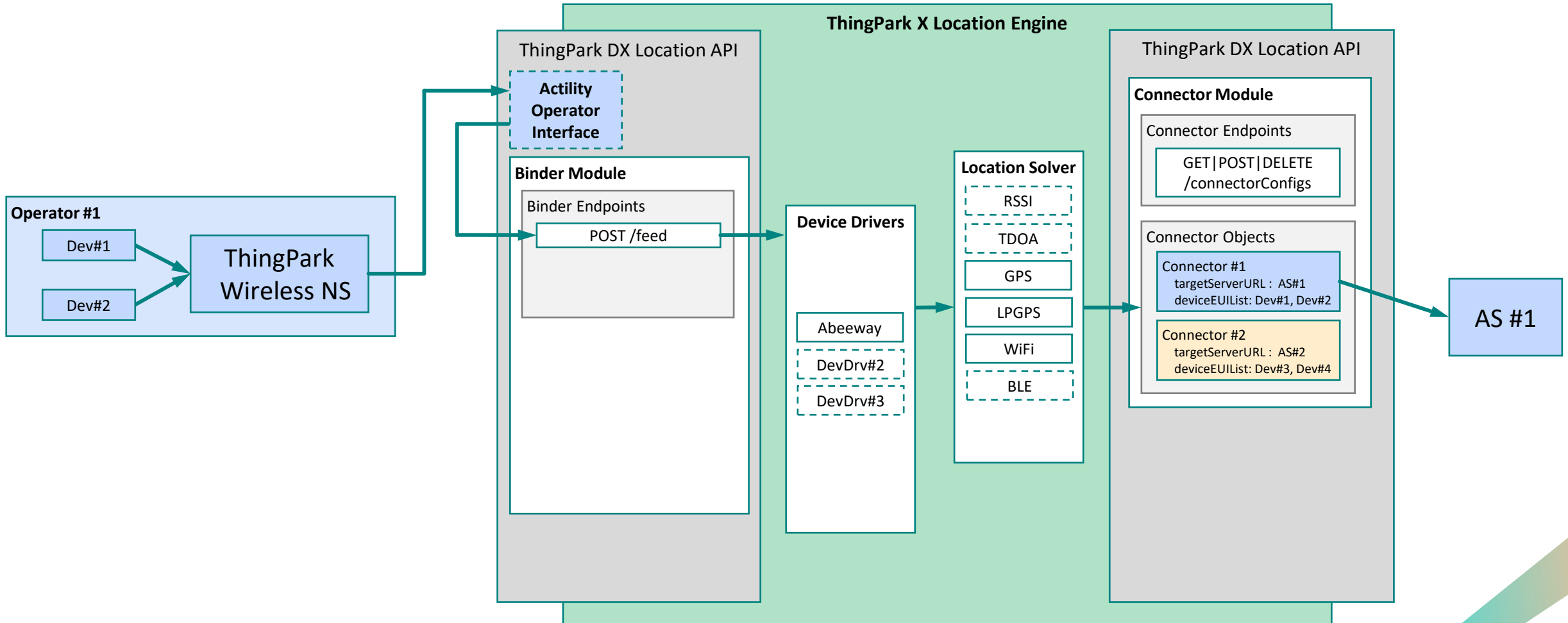
## Uplink Routing: Operator Interfaces

- Operator interfaces are API translators between different vendors' Network Servers and TPXLE /feed API.
- The Activity TPW Operator interface already exists.
- Operators with 3<sup>rd</sup> party NS will have to develop and maintain their own operator interface.



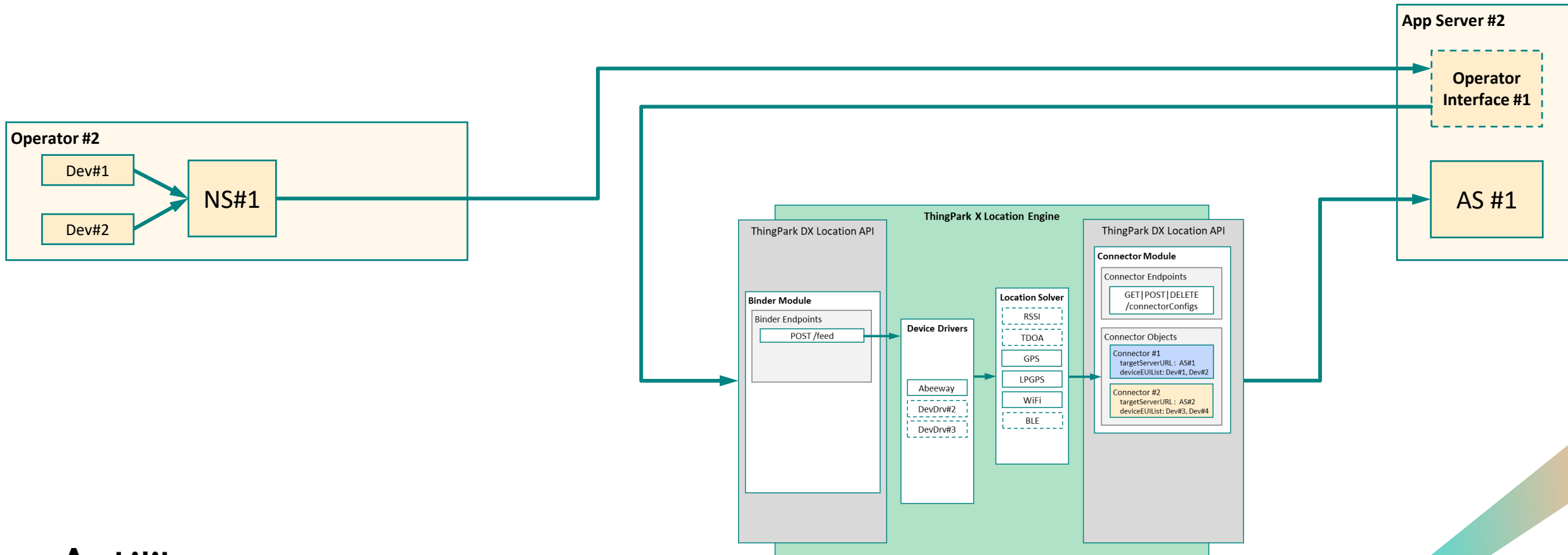
# ThingPark X Location Engine and DX Location API

## Uplink Routing: Activity Operator Interface



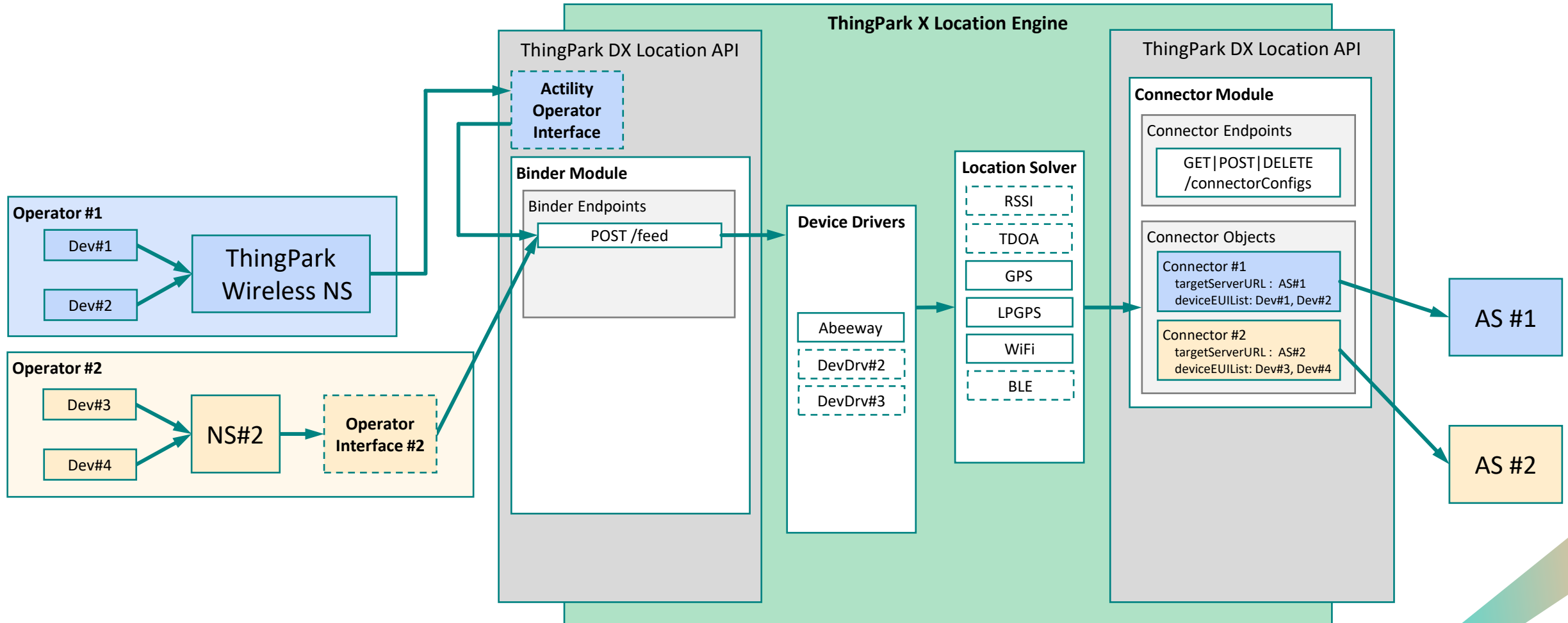
# ThingPark X Location Engine and DX Location API

## Uplink Routing: 3<sup>rd</sup> party operator interface in practice



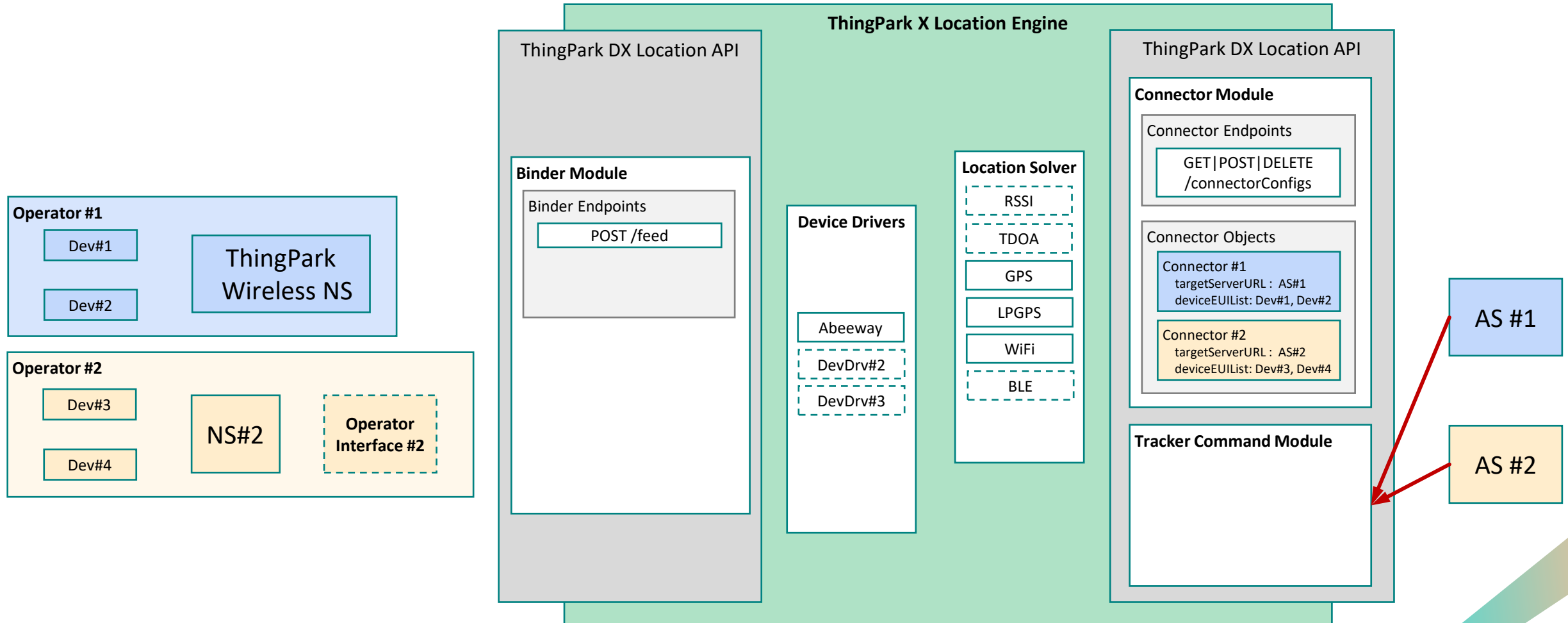
# ThingPark X Location Engine and DX Location API

## Uplink Routing: Full view



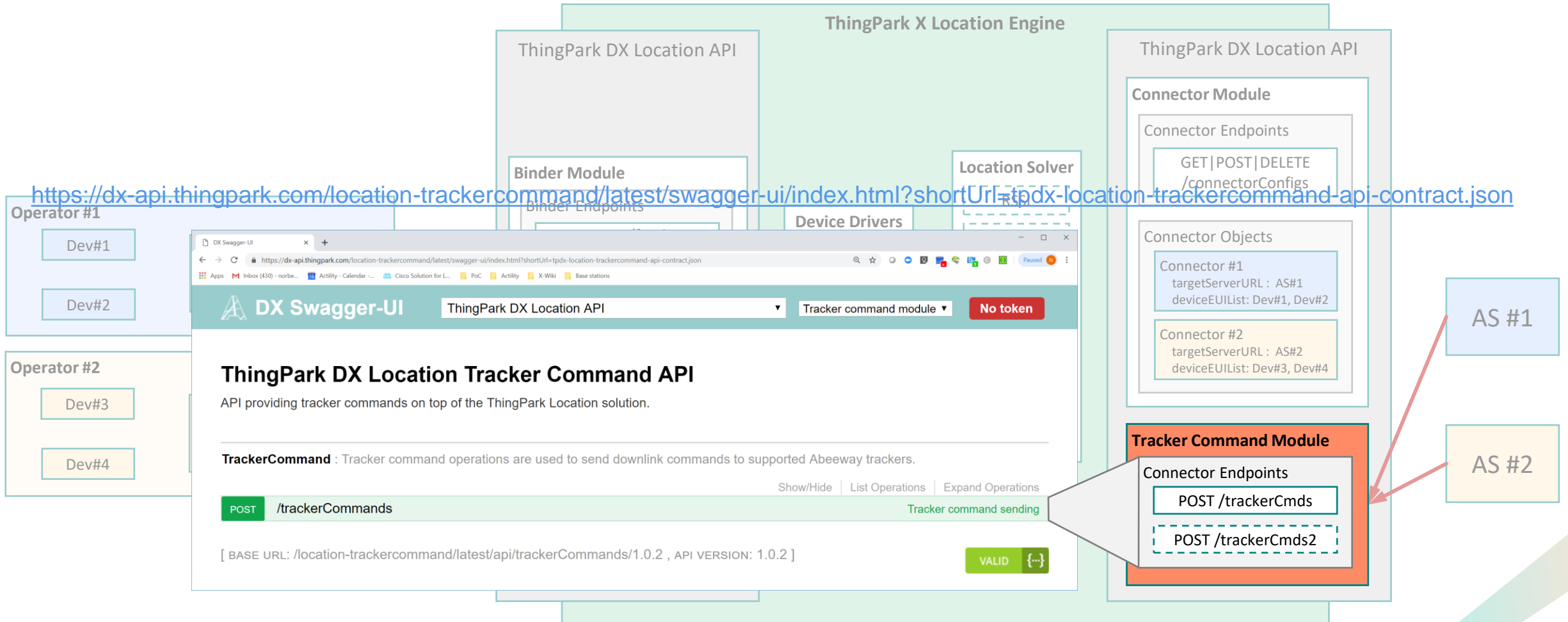
# ThingPark X Location Engine and DX Location API

## Downlink Routing: Tracker Command Module



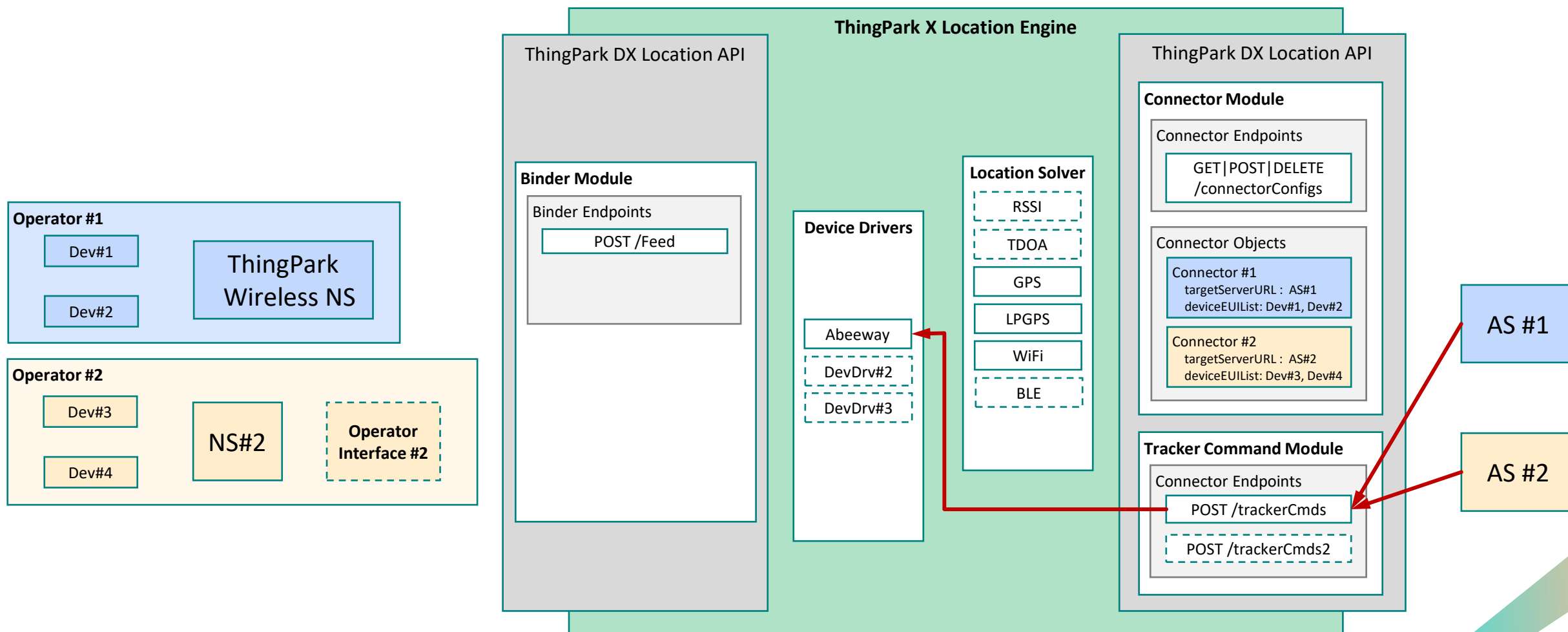
# ThingPark X Location Engine and DX Location API

## Downlink Routing: Tracker Command Module



# ThingPark X Location Engine and DX Location API

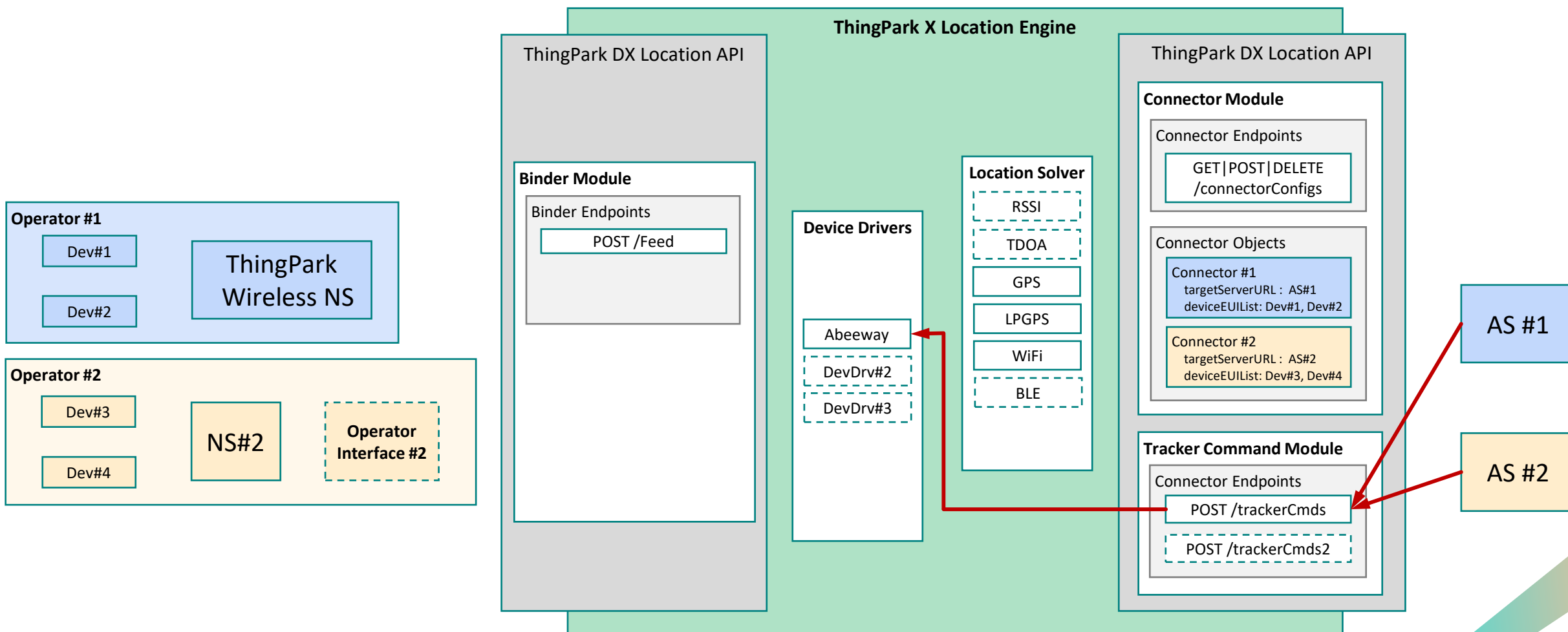
## Downlink Routing: Abeeway Driver (CODEC)





# ThingPark X Location Engine and DX Location API

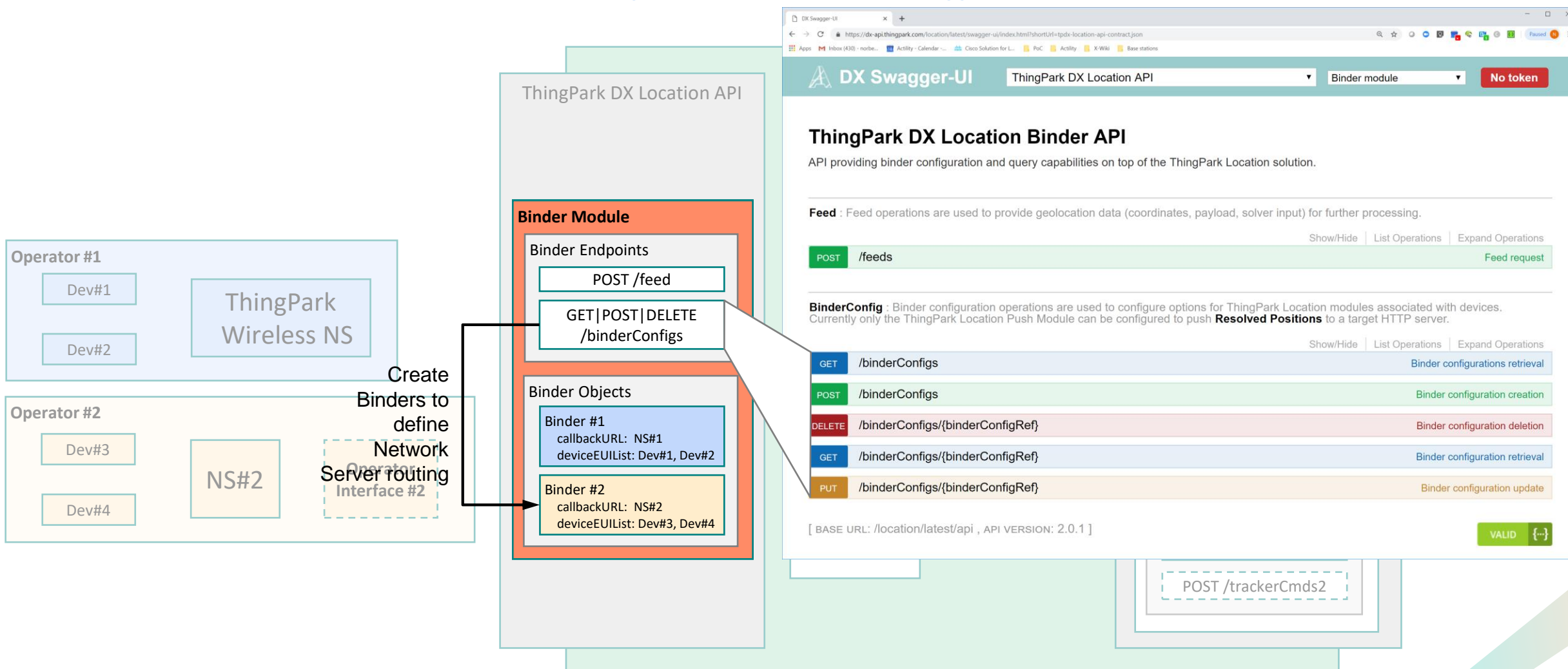
## Downlink Routing: Abeeway Driver (CODEC)



# ThingPark X Location Engine and DX Location API

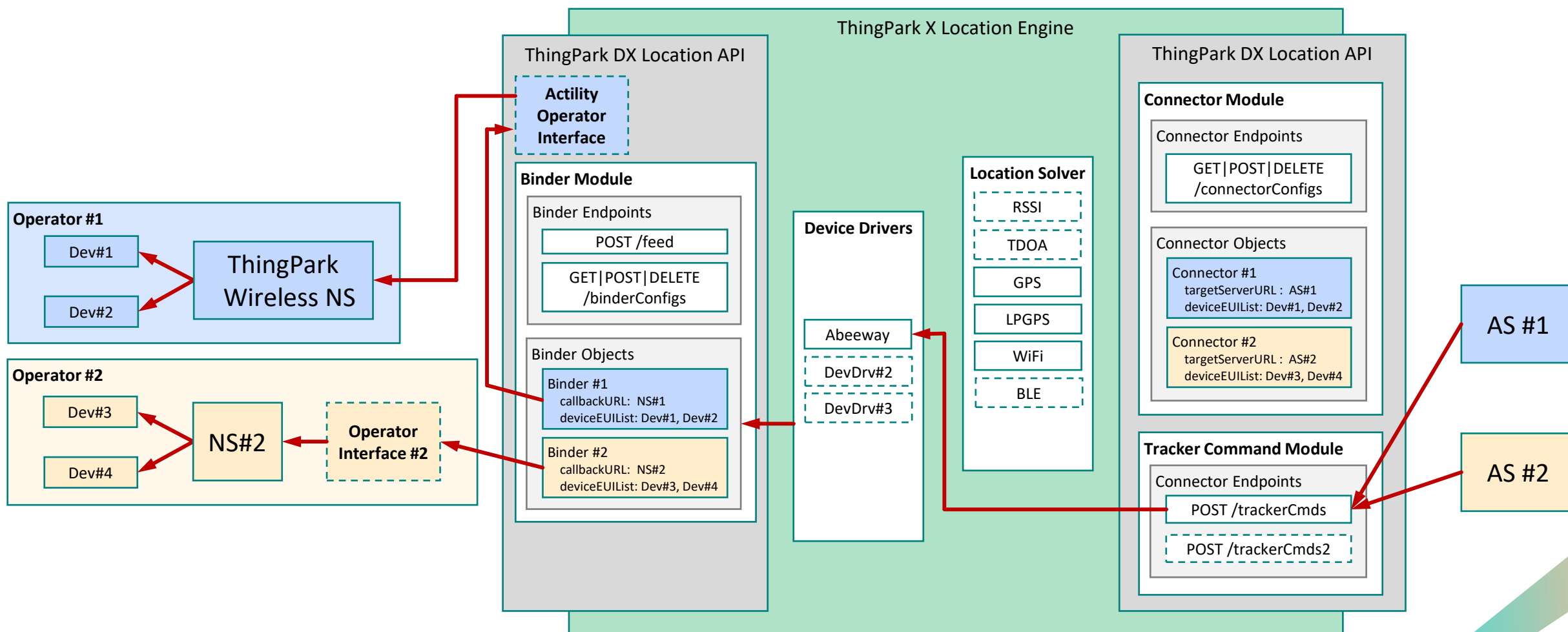
## Downlink Routing: Abeeway Driver (CODEC)

<https://dx-api.thingpark.com/location/latest/swagger-ui/index.html?shortUrl=tpdx-location-api-contract.json>



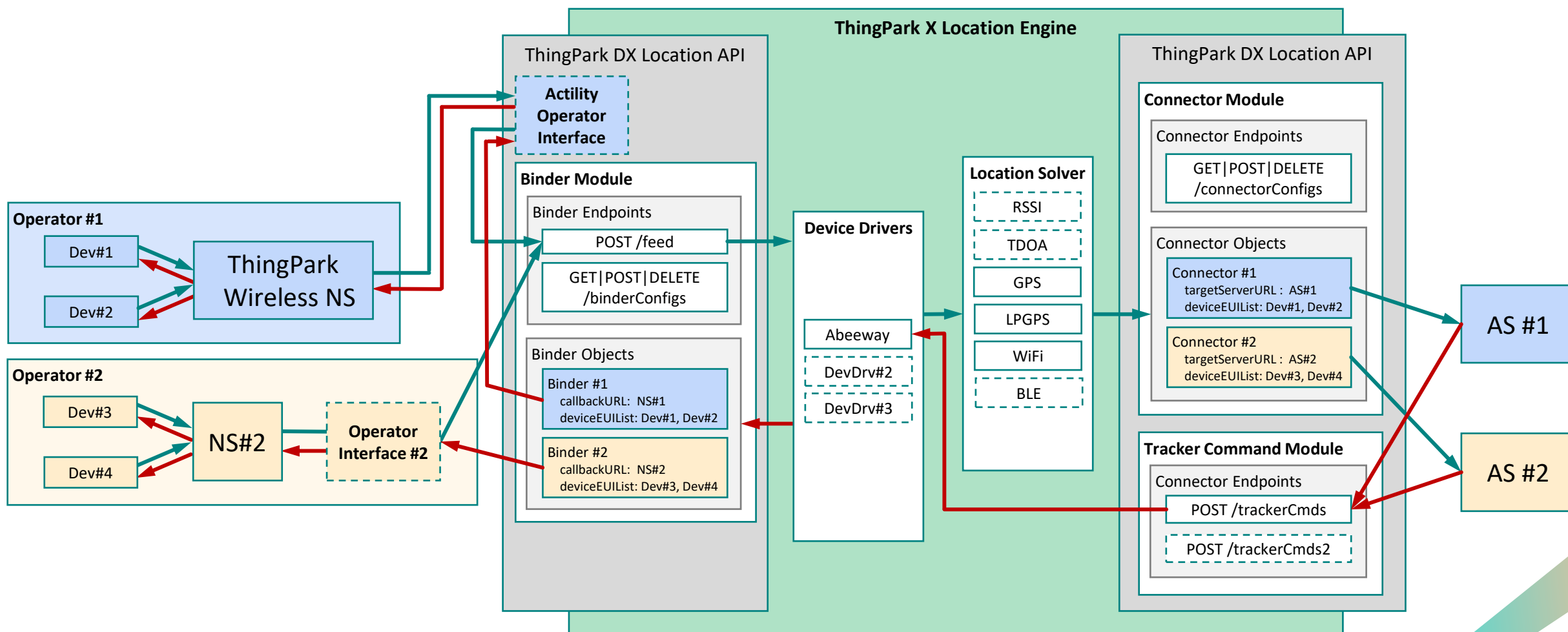
# ThingPark X Location Engine and DX Location API

## Downlink Routing: Full view

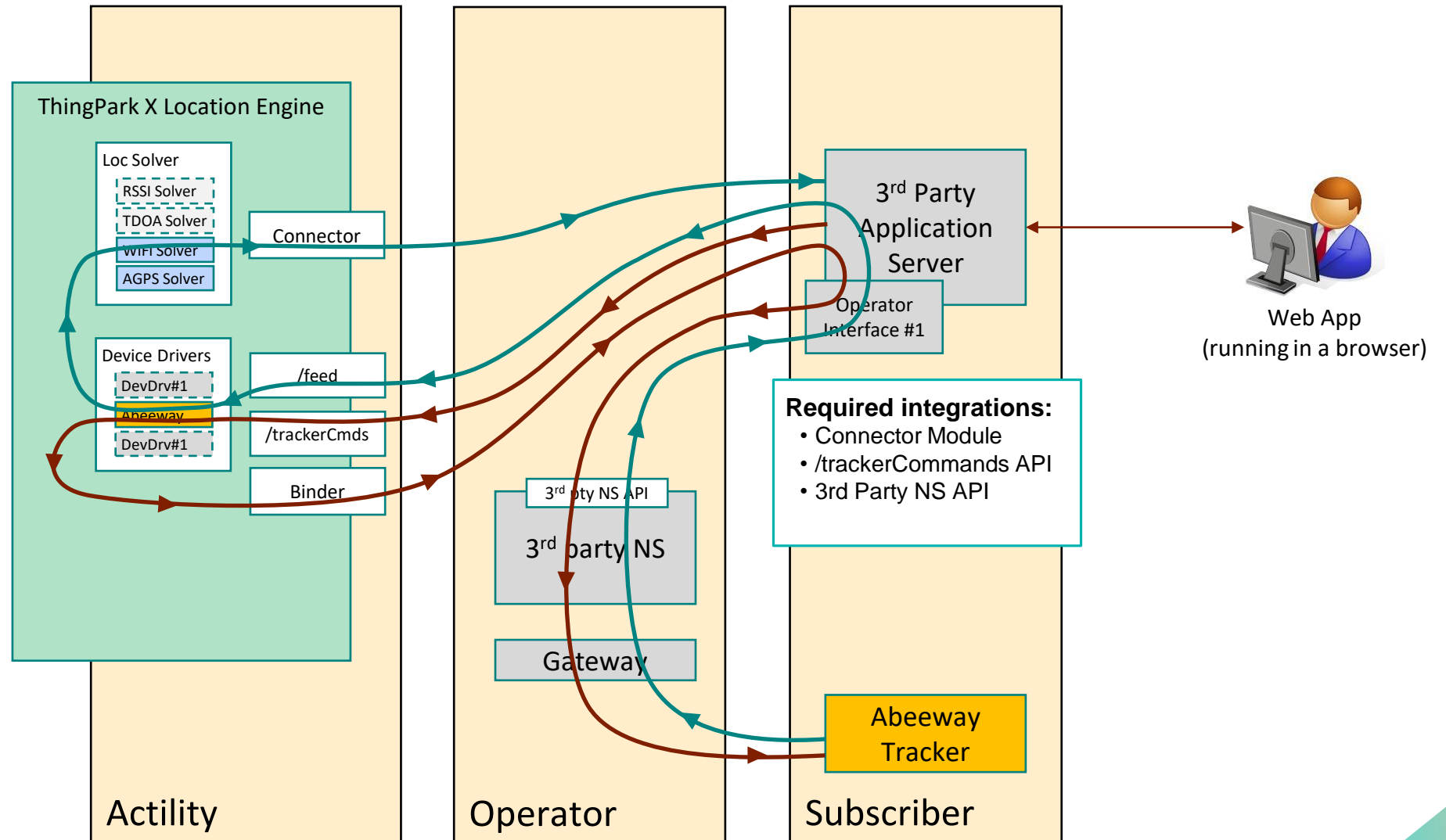


# ThingPark X Location Engine and DX Location API

## Uplink and Downlink Routing: Full view



# TPX LE Network Integration



# Agenda

- ThingPark X Location Engine message flows
- Hands-on exercise

Activity

# Exercise #1

Set up uplink data flow



Activity

# Exercise #1: Set up uplink data flow

---

1. Register for a free ThingPark Developer Account at <https://partners.thingpark.com/>
2. Generate an Access Token to be used for authorizing requests to DX Location API
  - Use the swagger-ui of the DX Admin API on the following link: <https://dx-api.thingpark.com/admin/latest/swagger-ui/index.html?shortUrl=tpdx-admin-api-contract.json>
  - Send a POST request to the /oauth/token endpoint with the following parameters in the swagger-ui form:
    - grant\_type=client\_credentials
    - client\_id=dev1-api/{{your thingpark userid}}
    - client\_secret={{your password}}
    - validityPeriod=infinite
  - Copy the access\_token filed from the response body and save it in a text file so that you can use it whenever you send new API requests



# Exercise #1: Set up uplink data flow

---

## 3. Create an operator interface within your AS that performs the following tasks:

- Receive uplink messages from the NS
- Convert the message body to the expected format of ThingPark X Location Engine's [/feed](#) endpoint
- Add the Access Token generated at point #2 to the Authorization header of the message body. The Authorization header must be added in the following form: "Authorization: Bearer {{your token}}"
- Forward the uplink message to ThingPark X Location Engine's [/feed](#) endpoint

An example operator interface app written in nodejs is available here:

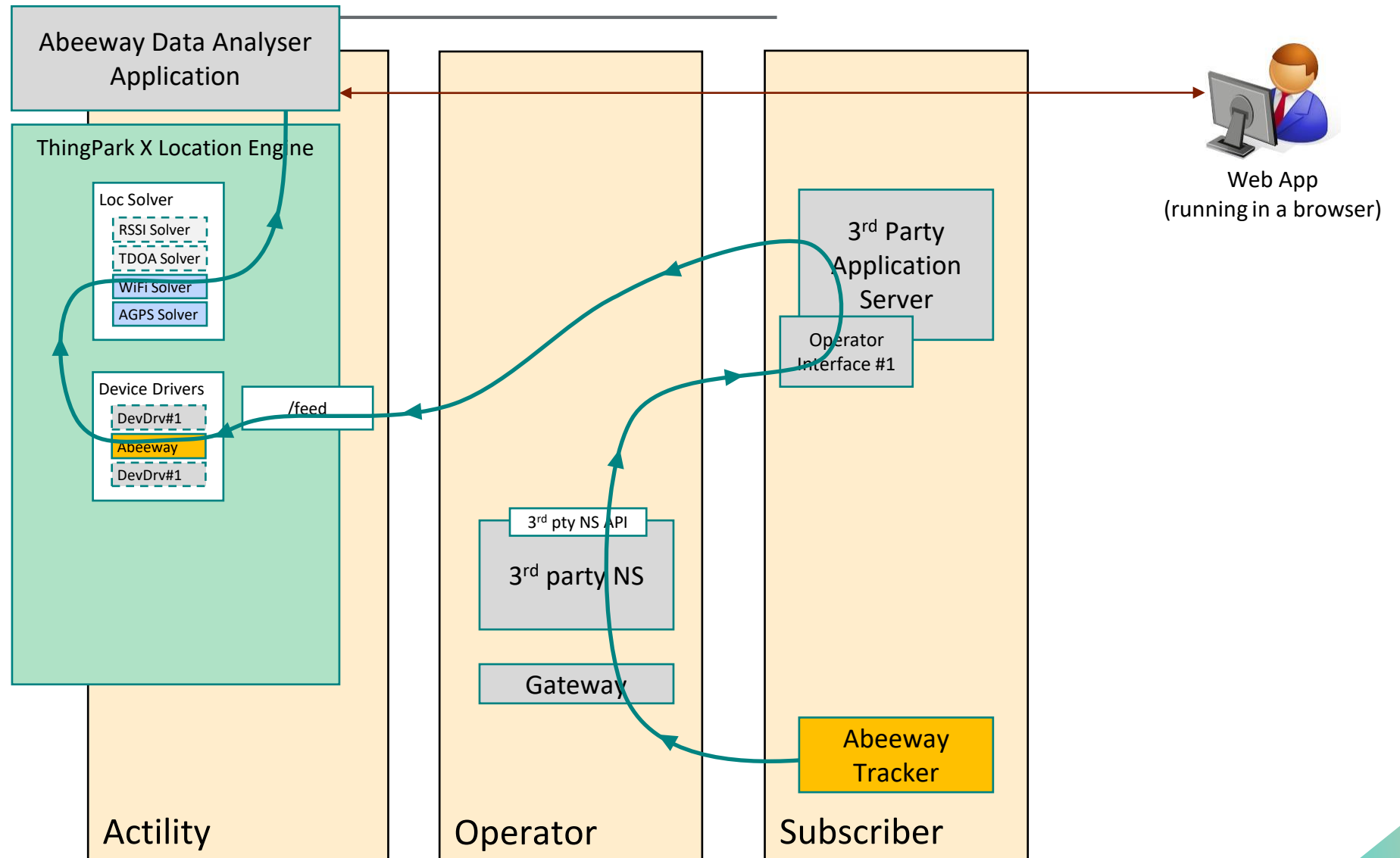
- <https://github.com/norbertherbert/tpxle-proxy.git>

## 4. Provision your devices on the NS and route all uplink messages from the NS to the operator interface of your AS

## 5. Log-in to the Abeeway Data Analyser App and check if you can see your devices on the map

- The url of the app is the following:  
<https://dev1.thingpark.com/abeewayDeviceAnalyzer/>
- The user credentials (userid and password) are the same as those that you got after registering at <https://partners.thingpark.com/>
- After you have logged in select your devices
- Then switch to the "Map" tab to see the trackers on the map.

# Exercise #1: Set up uplink data flow



# Exercise #2

Set up downlink data flow



## Exercise #2: Configuration steps

---

1. Perform all the steps of exercise #1
2. Set up a new binder through the DX Location API's binder module so that it sends all downlink messages to your Operator Interface of your AS
  - Go to the swagger-ui of DX Location API  
<https://dx-api.thingpark.com/location/latest/swagger-ui/index.html?shortUrl=tpdx-location-api-contract.json>
  - Make sure that the drop down fields at the top of the GUI are "DX Location API and "Binder Module"
  - Click on the red "No token" button at the top of the page and copy your token (that you created in exercise #1) prefixed with the word "Bearer " like this: "Bearer {{your token}}" then click on the "Set token button"
  - Send a POST request to the [/binderConfigs](#) endpoint after editing the message body so that the "callbackURL" points at the URL of your Operator Interface of your AS. Please also make sure that the "deviceEUIList" includes the Device EUIs of all your tracker devices.
3. Complement the functionalities of your Operator Interface with the following tasks:
  - Receive downlink messages from ThingPark X Location Engine
  - Convert the message body to the expected format of the API of the NS
  - Forward the downlink message to the NS

An example operator interface app written in nodejs is available here:

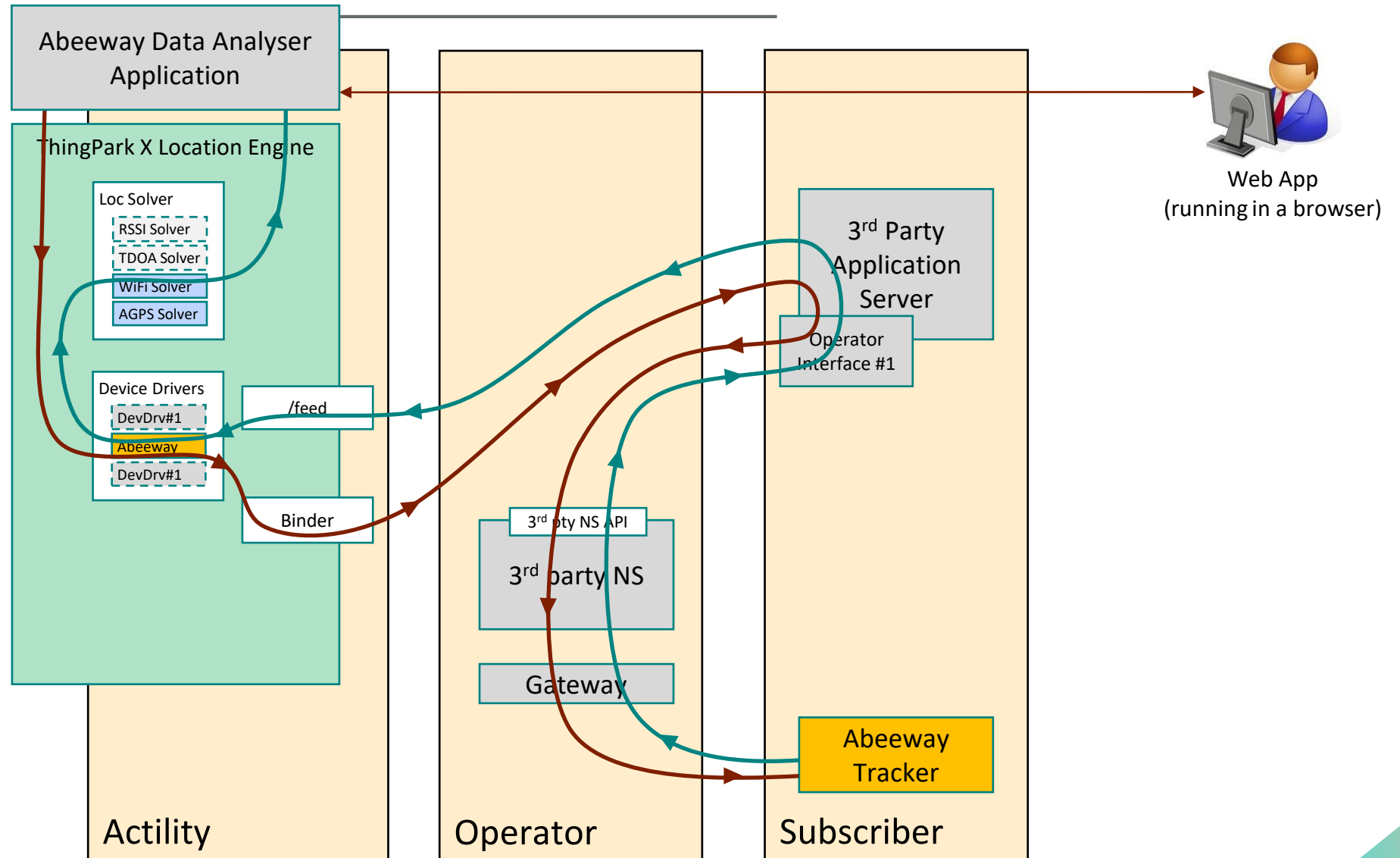
- <https://github.com/norbertherbert/tpxle-proxy.git>

## Exercise #2: Configuration steps

---

4. Log-in to the Abeeway Data Analyser App and send a tracker configuration message to the device. (e.g. you can change the tracking mode from GPS to WiFi)
  - Log in at the link below  
<https://dev1.thingpark.com/abeewayDeviceAnalyzer/>
  - Select your devices
  - Switch to the “Device configuration” tab and click the “Load default config” button.
  - Modify the configuration as you wish and click on the “Apply” button at the bottom.
5. Check in the logs of the NS that the downlink messages sent by the app arrive.
6. Check at the Abeeway Device Analyser App if the device changed tracking mode from GPS to WiFi.
  - Switch to the “Location log” tab and see latest messages sent by the device

## Exercise #2: Set up downlink data flow



# Exercise #3

Route uplink messages to the AS



Activity

## Exercise #3: Route uplink messages to the AS

---

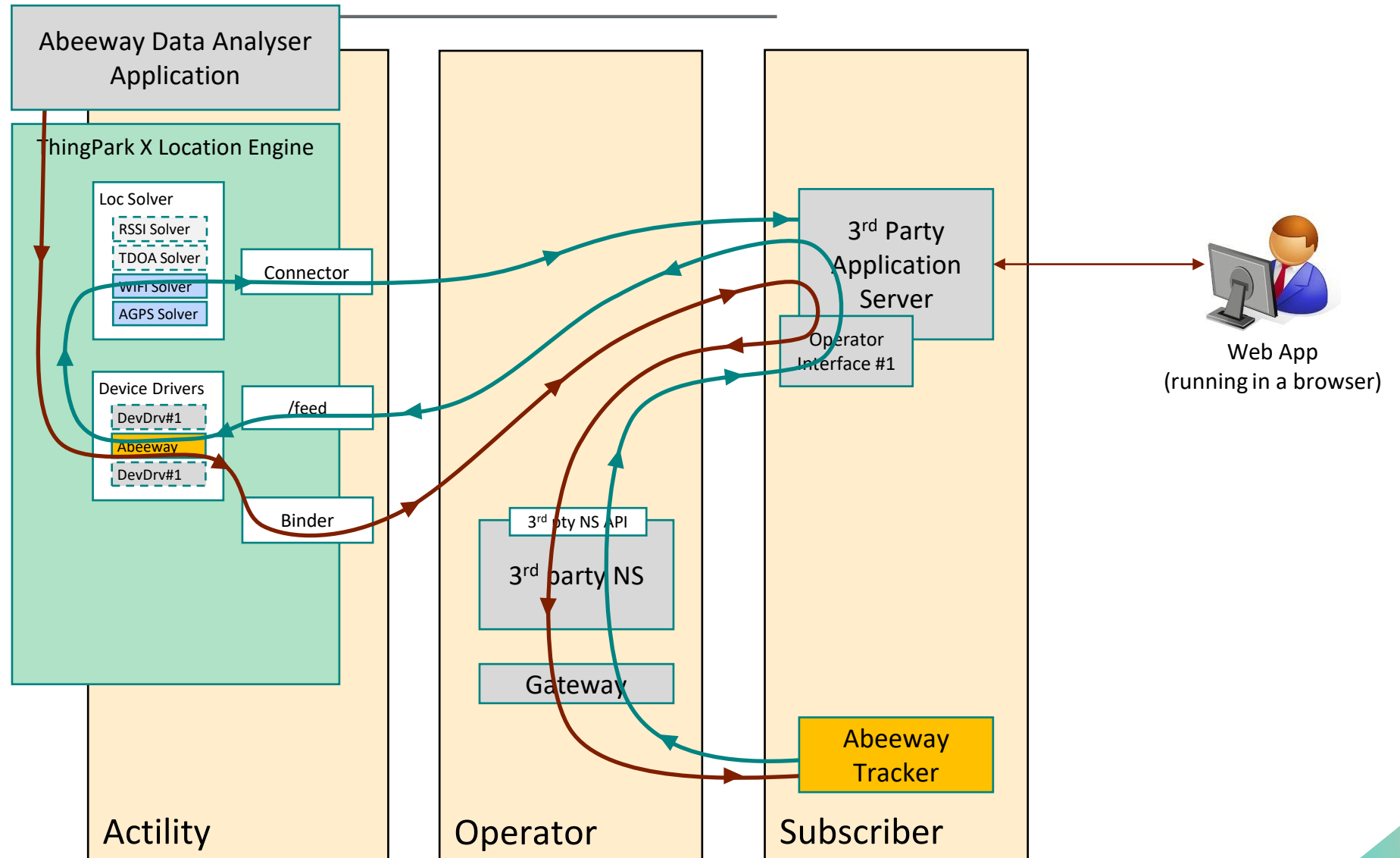
1. Perform all the steps of exercise #2
2. Set up a new connector through the DX Location API's connector module so that it sends all downlink messages to your AS
  - Go to the swagger-ui of DX Location API  
<https://dx-api.thingpark.com/location/latest/swagger-ui/index.html?shortUrl=tpdx-location-api-contract.json>
  - Make sure that the drop down fields at the top of the GUI are "DX Location API and "Connector Module"
  - If you haven't done yet, click on the red "No token" button at the top of the page and copy your token (that you created in exercise #1) prefixed with the word "Bearer " like this: "Bearer {{your token}}" then click on the "Set token button"
  - Send a POST request to the [/connectorConfigs](#) endpoint after editing the message body so that the "targetServerURL" points at the URL of your AS. Please also make sure that the "deviceEUIList" includes the Device EUIs of all your tracker devices.
3. Check at your AS if resolved uplink messages arrive properly.

The API of an example app written in nodejs is available here:

  - <https://github.com/norbertherbert/tpxle-proxy.git>



## Exercise #3: Route uplink messages to the AS



# Exercise #4

Send downlink messages from the AS



Activity

## Exercise #4: Send downlink messages from the AS

---

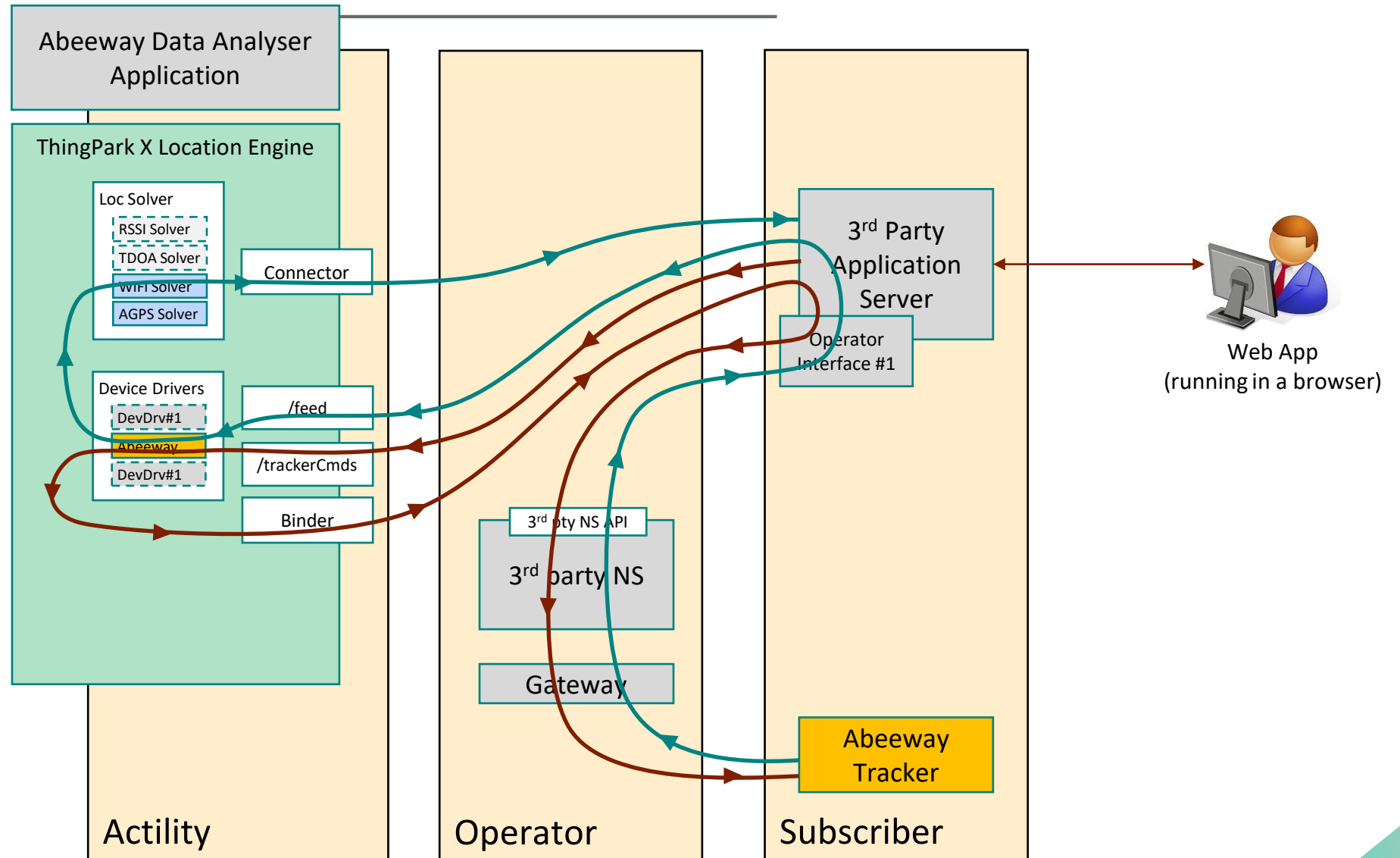
1. Perform all the steps of exercise #3
2. Configure your AS so that it sends downlink messages (tracker commands) to the /trackerCommands endpoint of DX Location API.
  - In order to see the /trackerCommands module go to the swagger-ui of DX Location API  
<https://dx-api.thingpark.com/location/latest/swagger-ui/index.html?shortUrl=tpdx-location-api-contract.json>
  - Make sure that the drop down fields at the top of the GUI are “DX Location API and “Tracker command module”

## Exercise #4: Send downlink messages from the AS

---

3. Send an example tracker command to the device (e.g. change the operation mode to Motion Tracking)
  - For testing purposes you can send a tracker command to the device manually for the AS
  - Go to the swagger-ui of DX Location API  
<https://dx-api.thingpark.com/location/latest/swagger-ui/index.html?shortUrl=tpdx-location-api-contract.json>
  - Make sure that the drop down fields at the top of the GUI are “DX Location API and “Tracker command module”
  - If you haven’t done yet, click on the red “No token” button at the top of the page and copy your token (that you created in exercise #1) prefixed with the word “Bearer “ like this: “Bearer {{your token}}” then click on the “Set token button”
  - Send a POST request to the [/trackerCommands](#) endpoint after editing the message body. (At least the “deviceEUIList” parameter has to be updated.
4. By analysing uplink message logs on the AS, verify if the tracker has changed its operation mode.

## Exercise #4: Send downlink messages from the AS





Summary

Actility