

# Application

## Notes and Ideas

The network interception I am thinking will be an array inside the application itself. The network will need a list of application ids it is associated with I think but probably not more than that.

### Requirements

- Should allow a single application to support multiple integrations

## API and Data Shape

### Create Application

*Note: integrations and networks is optional.*

POST `/api/applications`

### Body

```
{
  "name": "appName",
  "description": "appDescription",
  "payloadCodec": "string",
  "payloadDecoderScript": "string",
  "payloadEncoderScript": "string",
  "validationScript": "string",
  "supportsDownLink": "boolean",
  "running": "boolean",
  "applicationEUI": "string",
  "customerId": "string",
  "integrations": [
    {
      "baseUrl": "baseUrl",
      "reportingProtocolId": "reportingProtocolId"
    }
  ]
}
```

[Get Application](#)

```
GET /api/applications/:id
```

```
{
  "id": "string",
  "name": "appName",
  "description": "appDescription",
  "payloadCodec": "string",
  "payloadDecoderScript": "string",
  "payloadEncoderScript": "string",
  "validationScript": "string",
  "supportsDownLink": "boolean",
  "running": "boolean",
  "applicationEUI": "string",
  "customerId": "string",
  "integrations": [
    {
      "baseUrl": "baseUrl",
      "reportingProtocolId": "reportingProtocolId"
    }
  ],
  "networkDeployments": [
    {
      "networkId": "string",
      "remoteApplicationId": "string",

```

```
    "serviceProfileId": "string",
    "organizationId": "string",
    "securityData": {
        "accessToken": "string",
        "refreshToken": "string"
    }
}
]
```

## Get Many Applications

```
GET /api/applications?query=q
```

Returns 200

```
{
  "returned": "number",
  "available": "number",
  "results": [
    {
      "id": "string",
      "name": "appName",
      "description": "appDescription",
      "running": "boolean",
      "customerId": "string",
      "networkDeployments": ["networkId1", "networkId2"]
    }
  ]
}
```

## Update Application

```
PUT /api/applications/:id
```

Body

```
{
  "name": "appName",
  "description": "appDescription",
  "payloadCodec": "string",
  "payloadDecoderScript": "string",
  "payloadEncoderScript": "string",
  "validationScript": "string",
  "supportsDownLink": "boolean",
  "running": "boolean",
  "applicationEUI": "string",
  "customerId": "string",
  "integrations": [
    {
      "baseUrl": "baseUrl",
      "reportingProtocolId": "reportingProtocolId"
    }
  ],
  "networkDeployments": [
    "networkId1",
    "networkId2"
  ]
}
```

Return 200

```
{
  "id": "string",
  "name": "appName",
  "description": "appDescription",
  "payloadCodec": "string",
  "payloadDecoderScript": "string",
  "payloadEncoderScript": "string",
  "validationScript": "string",
  "supportsDownLink": "boolean",
  "running": "boolean",
  "applicationEUI": "string",
  "customerId": "string",
  "integrations": [
    {
      "baseUrl": "baseUrl",
```

```
        "reportingProtocolId": "reportingProtocolId"
      },
    ],
    "networkDeployments": [
      {
        "networkId": "string",
        "remoteApplicationId": "string",
        "serviceProfileId": "string",
        "organizationId": "string",
        "securityData": {
          "accessToken": "string",
          "refreshToken": "string"
        }
      }
    ]
  }
}
```

## Delete Application

```
DELETE /api/applications/:id
```

Body

```
{}
```

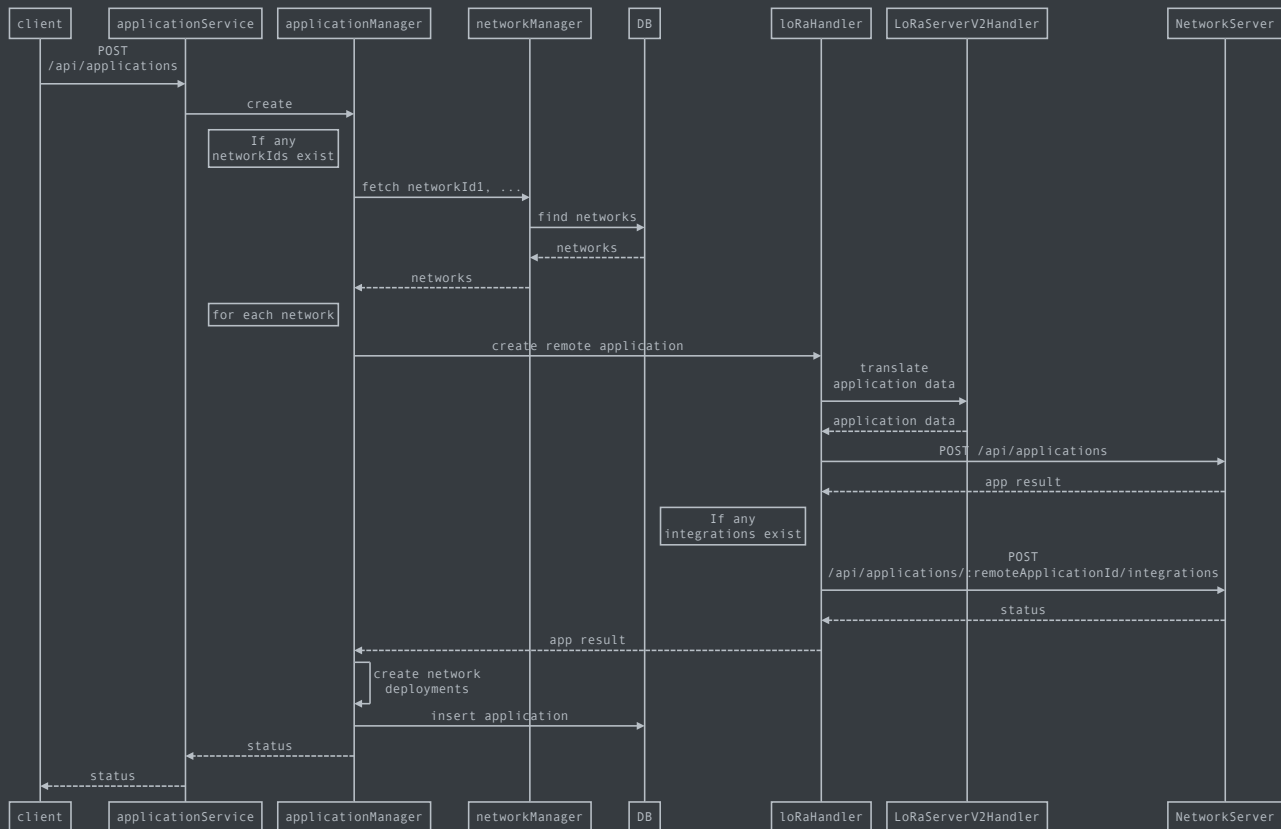
Return 204

```
{}
```

## Proposed Flow

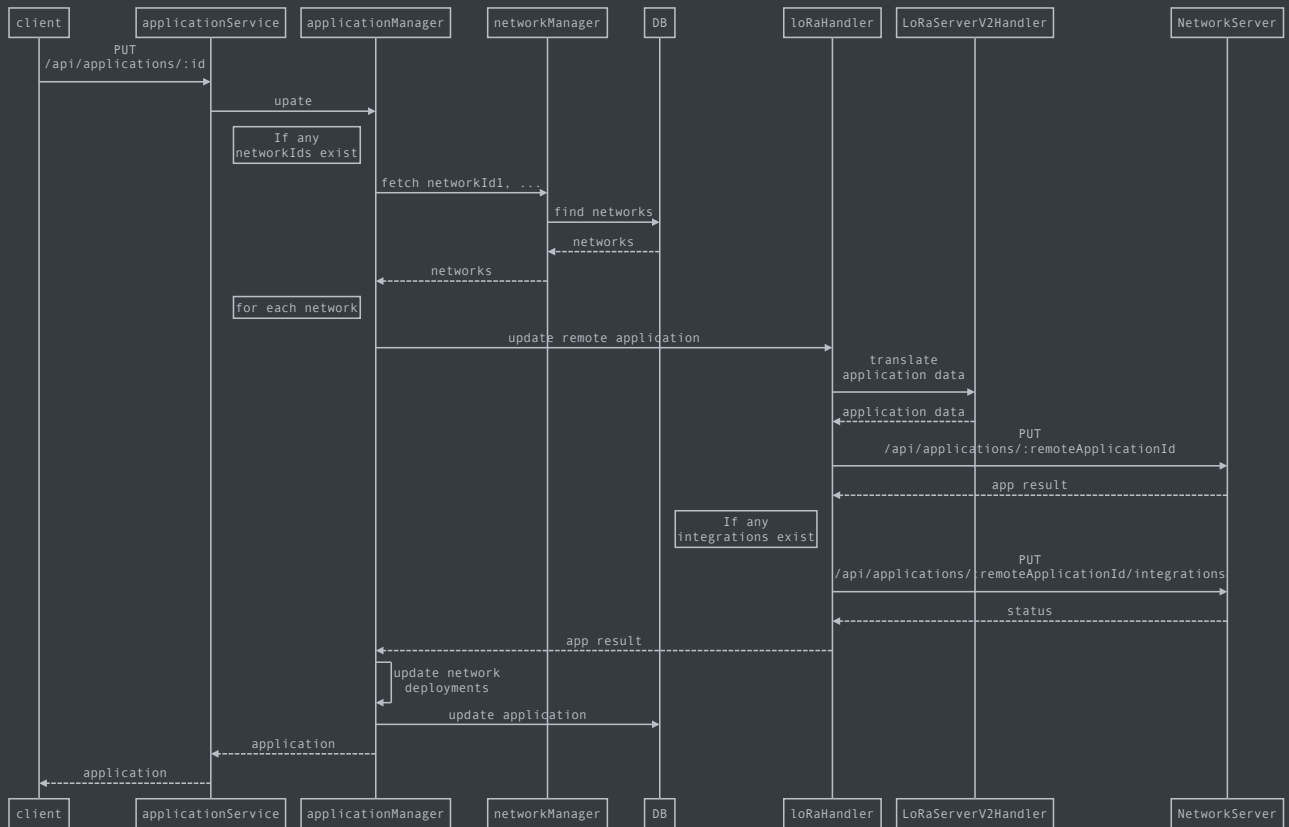
### New Application

*Note: Used LoRaServerV2 as an example, could be any protocol*

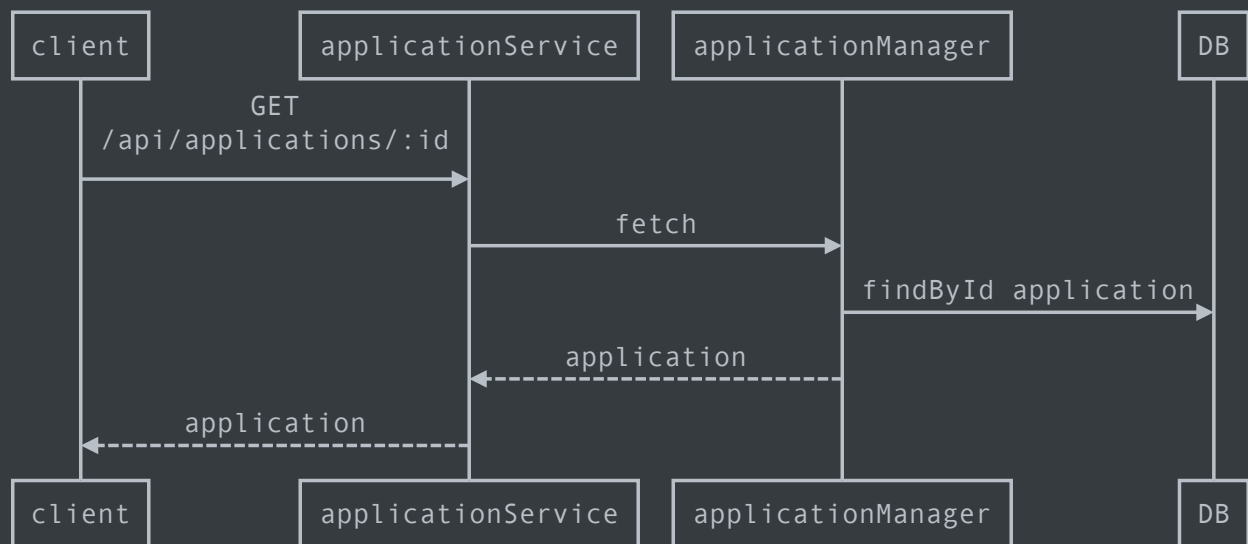


## Update Application

*Note: Used LoRaServerV2 as an example, could be any protocol*



## Get Application



## Delete Application

