

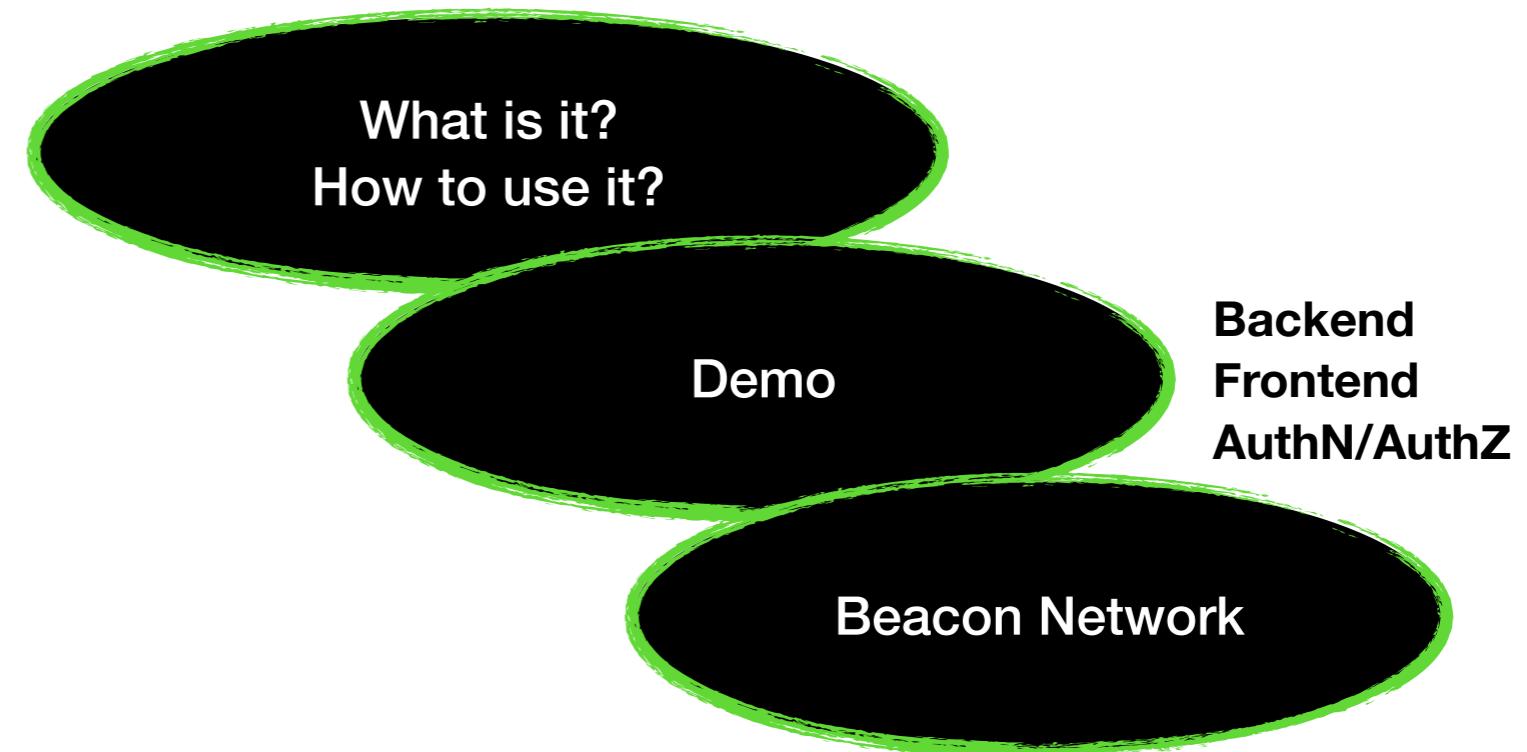
Beacon



Sabela de la Torre
Frédéric Haziza



Beacon



What is it? How to use it?

```
package org.ega_archive.elixirbeacon.service;

import java.util.List;

import org.ega_archive.elixirbeacon.dto.Beacon;
import org.ega_archive.elixirbeacon.dto.BeaconAlleleResponse;
import org.ega_archive.elixirbeacon.dto.BeaconRequest;
import org.ega_archive.elixirbeacon.enums.VariantType;
import org.ega_archive.elixircore.helper.CommonQuery;

import javassist.NotFoundException;

public interface ElixirBeaconService {

    /**
     * Returns the information about this beacon implementation and all datasets. It also specifies
     * the access type for each dataset:
     * <ul>
     * <li>PUBLIC: all.</li>
     * <li>REGISTERED: if the user is authenticated.</li>
     * <li>PROTECTED: if the user is authorized to access it.</li>
     *
     * @param commonQuery
     * @param referenceGenome
     * @return
     * @throws NotFoundException
     */
    public Beacon listDatasets(CommonQuery commonQuery, String referenceGenome) throws NotFoundException;

    /**
     * Executes the query against the beacon and basically answers yes or no.
     *
     * @param start
     * @param startMin
     * @param startMax
     * @param end
     * @param endMin
     * @param endMax
     * @param chromosome
     * @param referenceBases
     * @param alternateBases
     * @param datasetStableIds
     * @param referenceGenome
     * @param includeDatasetResponses
     * @return
     */
    public BeaconAlleleResponse queryBeacon(List<String> datasetStableIds, String variantType,
                                            String alternateBases, String referenceBases, String chromosome, Integer start,
                                            Integer startMin, Integer startMax, Integer end, Integer endMin, Integer endMax,
                                            String referenceGenome, String includeDatasetResponses);

    /**
     * Verifies that mandatory parameters are present and that all parameters are valid.
     */
}
```

Code



DB

What is it?
How to use it?

```
package org.neo4j.arktive.elasticsearch.beacon.services;
import java.util.List;
import org.neo4j.arktive.elasticsearch.beacon.Beacon;
import org.neo4j.arktive.elasticsearch.beacon.BeaconIndexer;
import org.neo4j.arktive.elasticsearch.beacon.BeaconRequest;
import org.neo4j.arktive.elasticsearch.beacon.BeaconService;
import org.neo4j.arktive.elasticsearch.helper.Commodity;
import org.elasticsearch.action.ActionListener;
public interface ElasticsearchService {
    /**
     * Executes the query against this beacon implementation and all datasets. It also specifies
     * the security roles for each dataset.
     */
    void executeQuery(String query, String[] roles, ActionListener<List<Beacon>> listener);
    /**
     * Returns true if the user is authenticated.
     */
    boolean isAuthenticated();
    /**
     * Returns true if the user is authorized to access the given role.
     */
    boolean isAuthorized(String role);
    /**
     * Gets the reference.
     */
    Commodity getReference();
    /**
     * Gets the referenceId.
     */
    String getReferenceId();
    /**
     * Gets the datasets.
     */
    List<String> getDatasets();
    /**
     * Gets the datasetsStale.
     */
    boolean getDatasetsStale();
    /**
     * Gets the datasetsResponses.
     */
    List<String> getDatasetsResponses();
    /**
     * Executes the query against the beacon and basically answers yes or no.
     */
    boolean start();
    boolean stop();
    boolean restart();
    boolean shutdown();
    boolean ready();
    boolean reference();
    boolean referenceDatasets();
    boolean datasetsStale();
    boolean datasetsResponses();
    boolean datasetsResponses();
    public BeaconIndexer queryBeaconList(String datasetStableId, String variantType,
                                         String referenceName, Integer startMin, Integer startMax, Integer endMin, Integer endMax,
                                         String referenceName, String datasetsStableId);
    /**
     * Verifies that mandatory parameters are present and that all parameters are valid.
     */
}
```

Code



DB

Demo

Beacon N

What is it?
How to use it?



What is it?
How to use it?



Do you have:

- ◆ a mutation from G to A
- ◆ on chromosome 12
- ◆ for genome GRCh37
- ◆ at position 2655179 ?



Eh?... No hablo ingle...

What is it?
How to use it?



Do you have:

- ◆ a mutation from G to A
- ◆ on chromosome 12
- ◆ for genome GRCh37
- ◆ at position 2655179 ?



query: http



Yes 👍 / No 👎 (in JSON)

What is it?
How to use it?



Do you have:

- ◆ a mutation from **G** to **A**
- ◆ on chromosome **12**
- ◆ for genome **GRCh37**
- ◆ at position **2655179** ?



query: http



Yes / No (in JSON)

What is it?
How to use it?



http://beacon/ query ?

assemblyId = GRCh37
& referenceName = 12
& referenceBases = G
& alternateBases = A
& start = 2655179



What is it?
How to use it?



http://beacon/ query ?

assemblyId = **GRCh37**
& referenceName = **12**
& referenceBases = **G**
& alternateBases = **A**
& start = **2655179**



What is it?
How to use it?



http://beacon/ query ?

assemblyId = **GRCh37**
& referenceName = **12**
& referenceBases = **G**
& alternateBases = **A**
& start = **2655179**

Basic query:

Y/N + a bit more



What is it?
How to use it?



http://beacon/ query ?

assemblyId = **GRCh37**
& referenceName = **12**
& referenceBases = **G**
& alternateBases = **A**
& start = **2655179**



Basic query:

Y/N + a bit more

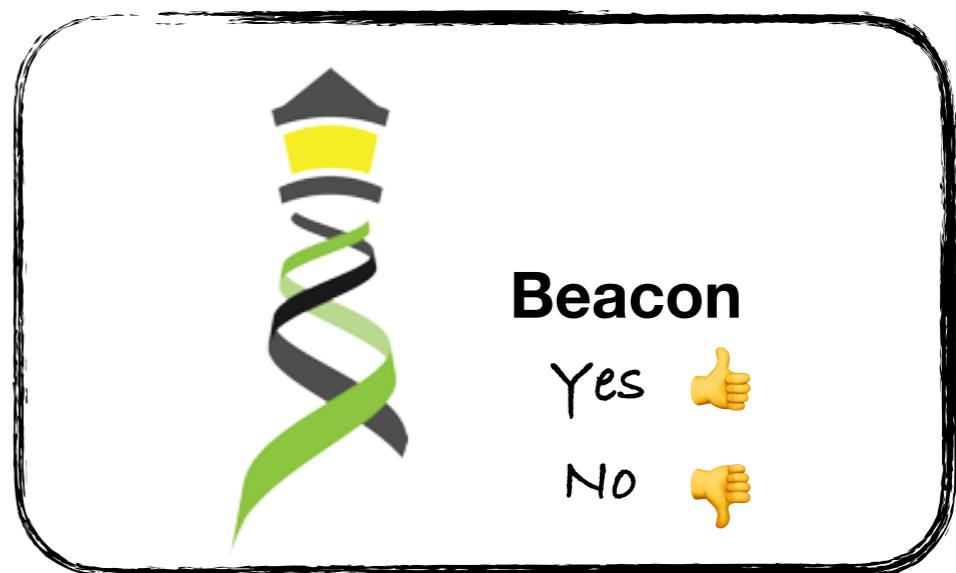
What is it?
How to use it?



Basic query: [http ://beacon/](http://beacon/) query ? parameters

- **assemblyId**
- **referenceName**
- **referenceBases**
- **alternateBases**
- **start (startMin / startMax)**
- **end (endMin / endMax)**

- **datasetIds**
- **includeDatasetResponses**
- **variantType**
(eg DUP, DEL, INS, INV, CNV, SNP, MNP)



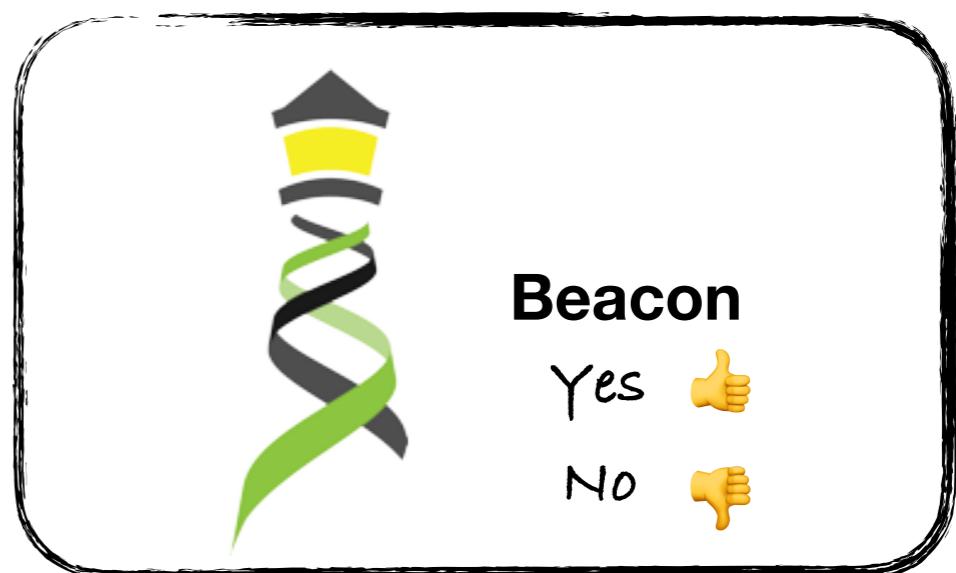
What is it?
How to use it?



Basic query: [http ://beacon/](http://beacon/) query ? parameters

- **assemblyId**
- **referenceName**
- **referenceBases**
- **alternateBases**
- **start (startMin / startMax)**
- **end (endMin / endMax)**

- **datasetIds**
- **includeDatasetResponses**
- **variantType**
(eg DUP, DEL, INS, INV, CNV, SNP, MNP)



+ variantCount
callCount
sampleCount

a bit more

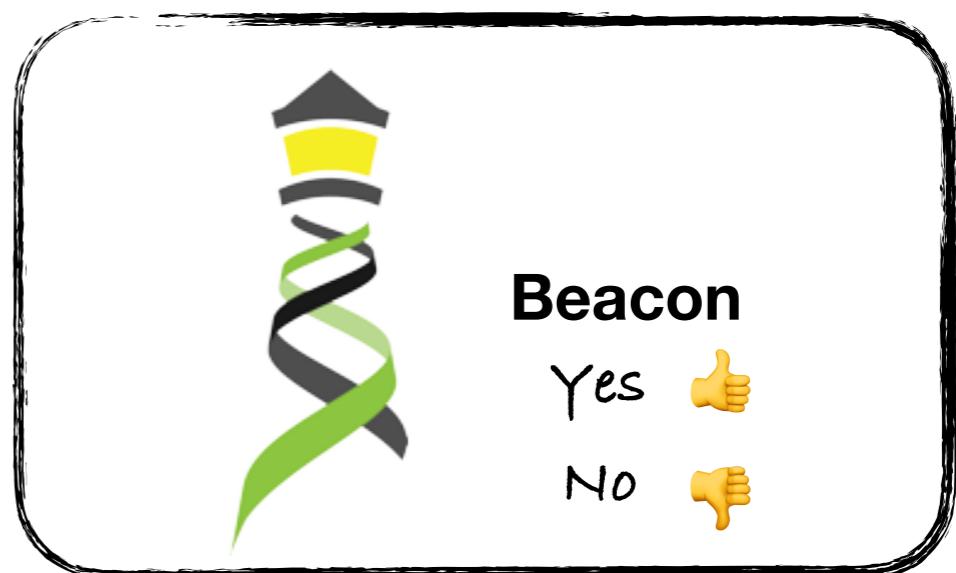
What is it?
How to use it?



Basic query: [http ://beacon/](http://beacon/) query ? parameters

- **assemblyId**
- **referenceName**
- **referenceBases**
- **alternateBases**
- **start (startMin / startMax)**
- **end (endMin / endMax)**

Specification



- **datasetIds**
- **includeDatasetResponses**
- **variantType**
(eg DUP, DEL, INS, INV, CNV, SNP, MNP)

+ variantCount
callCount
sampleCount

a bit more

What is it?
How to use it?

Specification

The screenshot shows the Swagger Editor interface with the title "GA4GH Beacon API Specification". The left side displays the API specification in YAML format, and the right side shows the generated API documentation.

API Specification (YAML):

```
openapi: 3.0.0
servers: []
info:
  version: "1.0.1"
  title: GA4GH Beacon API Specification
  description: >-
    A Beacon is a web service for genetic data sharing that can be
    queried for information about specific alleles.
  contact:
    email: beacon@google.org
  license:
    name: Apache 2.0
    url: 'http://www.apache.org/licenses/LICENSE-2.0.html'
externalDocs:
  description: 'Beacon Project'
  url: 'http://beacon-project.io/'
paths:
  /:
    get:
      description: Get information about the beacon
      operationId: getBeacon
      responses:
        '200':
          description: successful operation
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Beacon'
  /query:
    get:
      description: Get response to a beacon query for allele
      operationId: getBeaconAlleleResponse
      parameters:
        - name: referenceName
          description: 'Reference name (chromosome), Accession ID or
          assembly name (e.g., GRCh38).'


```

GA4GH Beacon API Specification

1.0.1 OAS3

<https://raw.githubusercontent.com/ga4gh-beacon/specification/master/beacon.yaml>

A Beacon is a web service for genetic data sharing that can be queried for information about specific alleles.

Contact the developer
Apache 2.0
Beacon Project

default

GET /

GET /query

POST /query

Schemas

What is it?
How to use it?

Beacon M

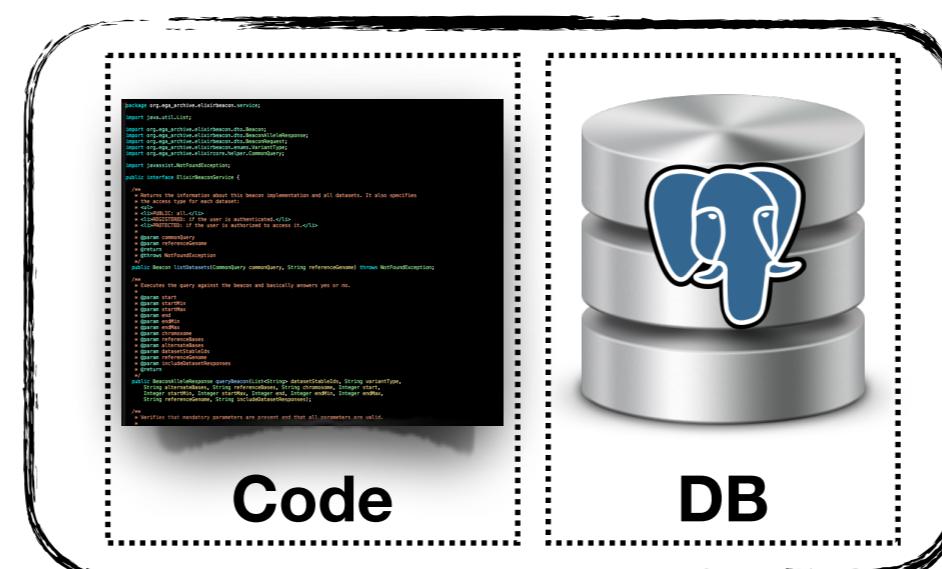
Demo



What is it? How to use it?

Beacon N

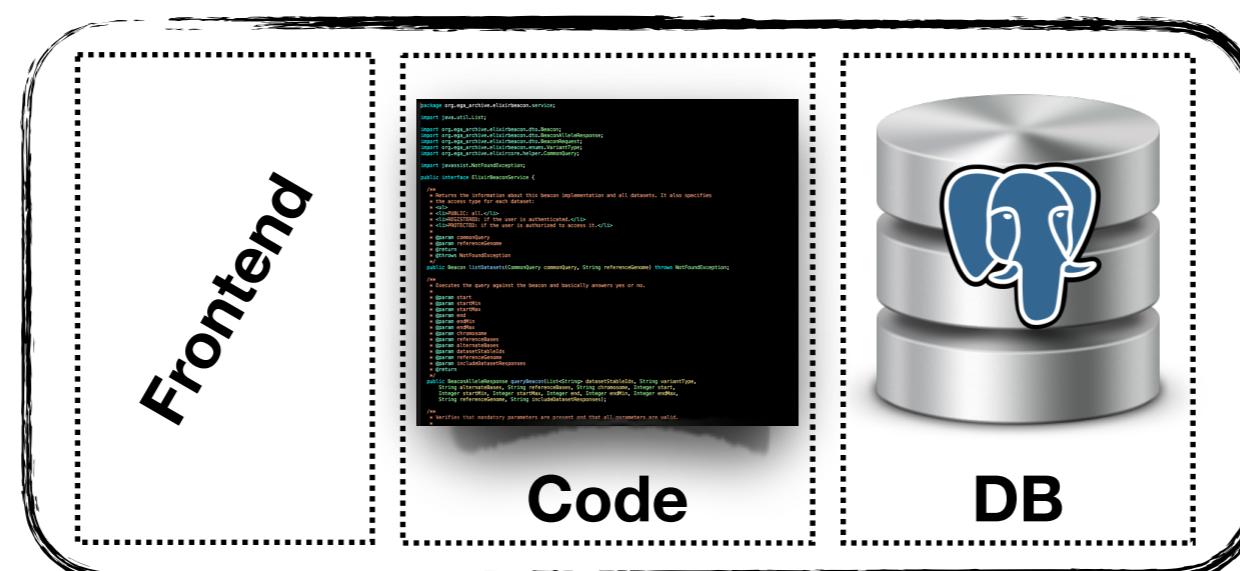
Demo



What is it? How to use it?

Beacon N

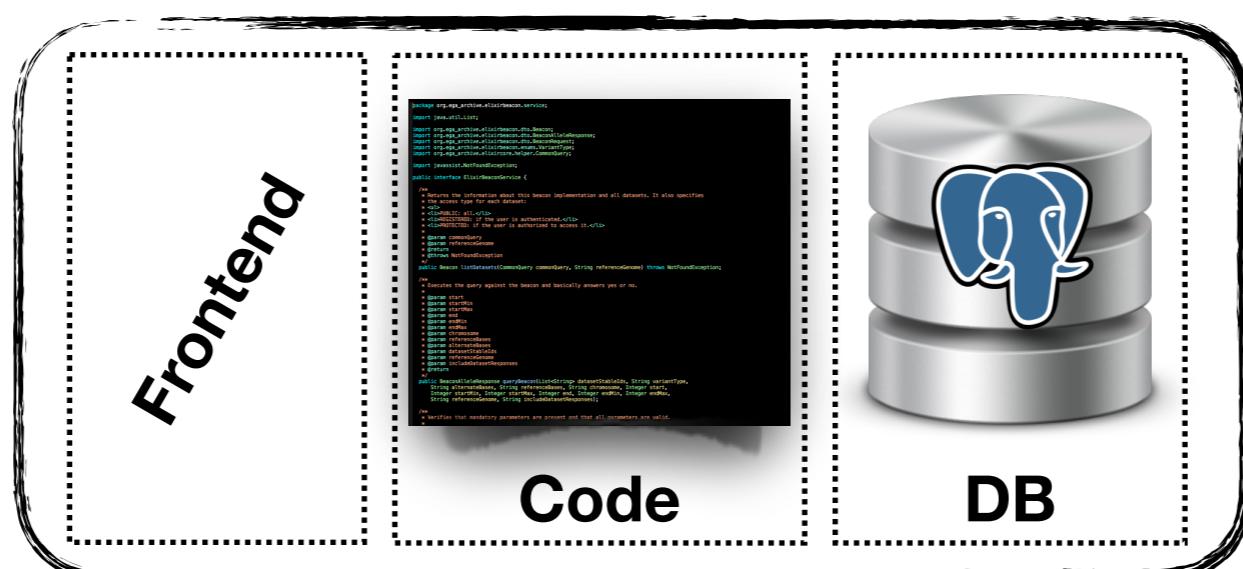
Demo



What is it?
How to use it?

Beacon M

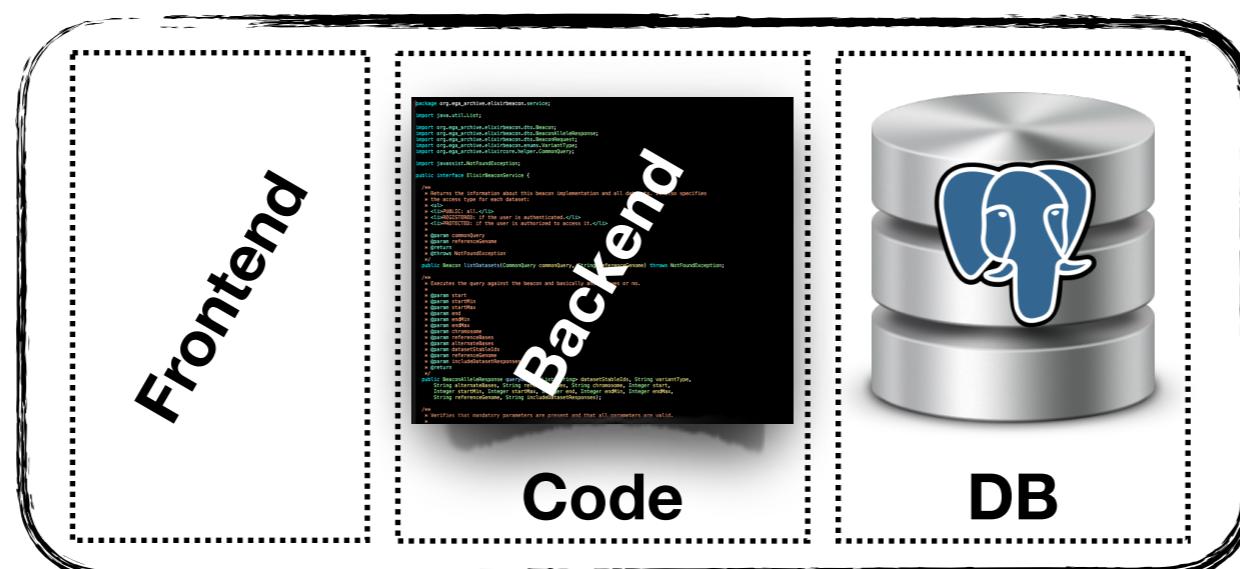
Demo



What is it?
How to use it?

Beacon M

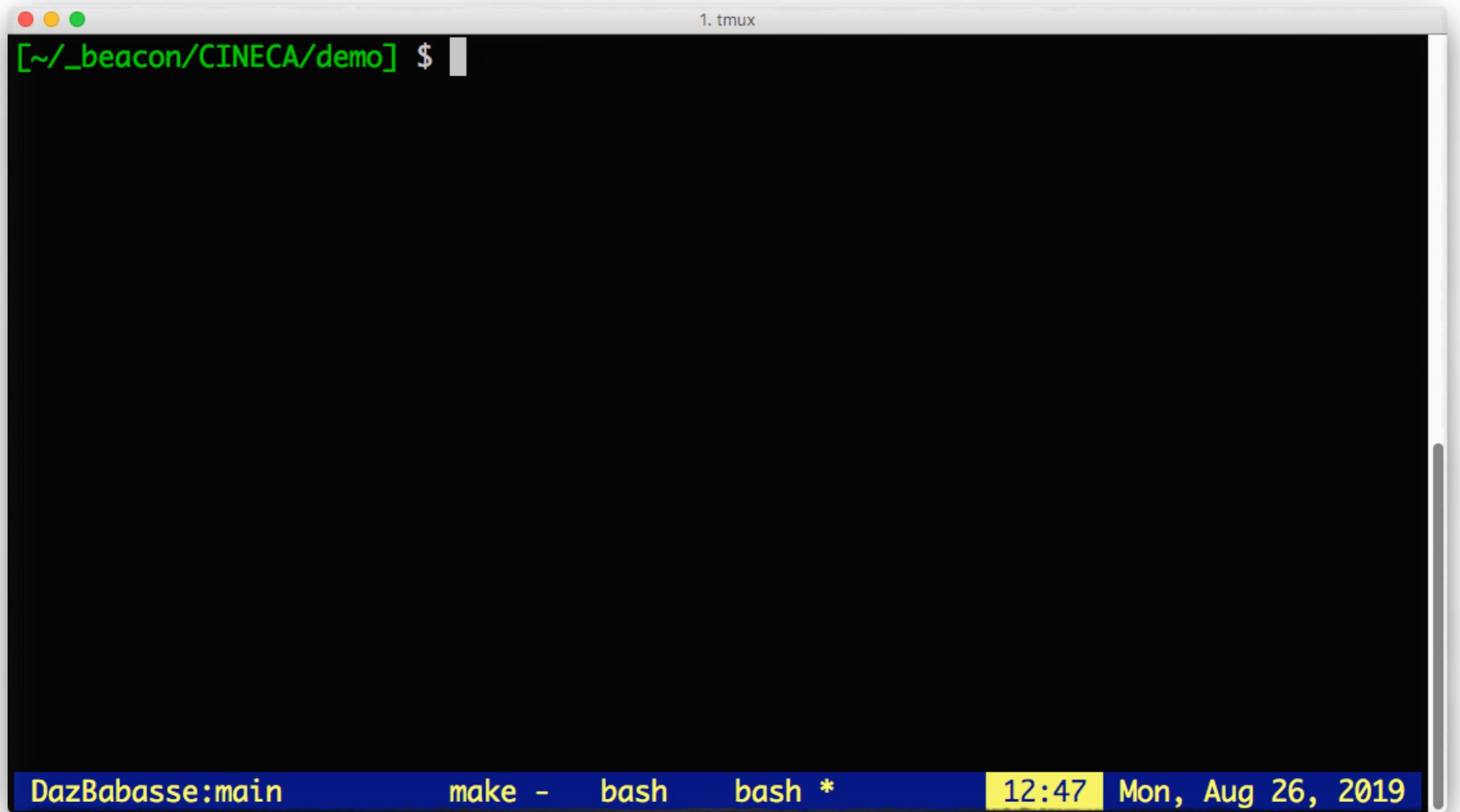
Demo



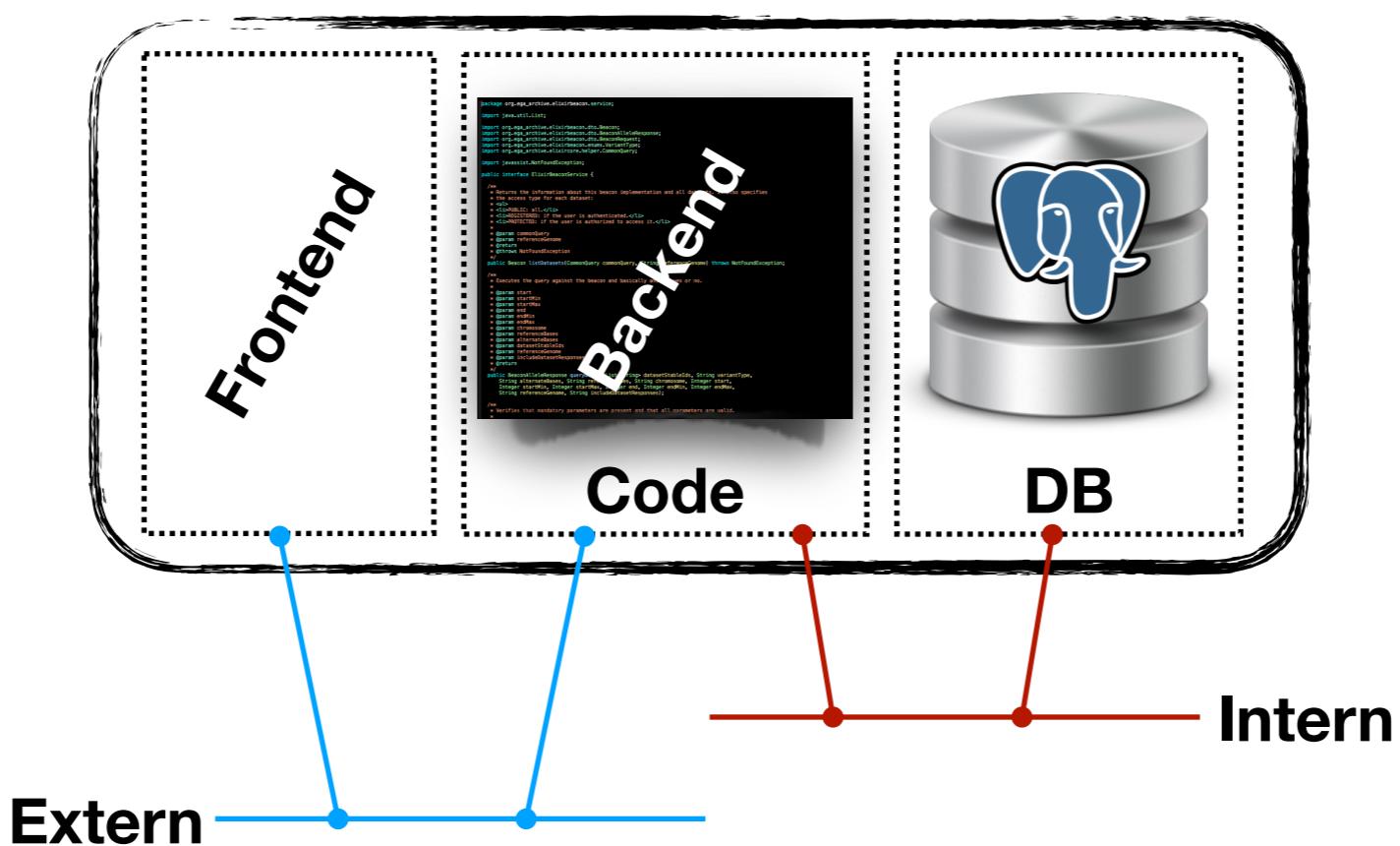
What is it?
How to use it?

Beacon N

Demo



Demo



What is it?
How to use it?

Demo

1. tmux

```
[~/_beacon/CINECA/demo] $ make dbshell
psql (11.4, server 11.3)
Type "help" for help.

cineca=# \dv
          List of relations
 Schema |      Name      | Type | Owner
-----+-----+-----+
 public | beacon_data_summary | view | cineca_admin
 public | beacon_dataset    | view | cineca_admin
 public | beacon_dataset_consent_code | view | cineca_admin
(3 rows)

cineca=# select id, stable_id from beacon_dataset;
 id |      stable_id
----+
 1 | 1000genomes
 2 | urn:hg:example-registered
 3 | urn:hg:example-controlled
(3 rows)

cineca=# select id, stable_id, description from beacon_dataset;
 id |      stable_id |                                description
----+-----+-----+
 1 | 1000genomes | Subset of variants of chromosomes 22 and Y from the 1000 genomes project
 2 | urn:hg:example-registered | Registered Dataset 1 with fake data (for Bona Fide researchers)
 3 | urn:hg:example-controlled | Controlled Dataset 2 with fake data
(3 rows)

cineca=#
```

Demo

http://beacon / query ?parameters.....

```
{  
    "apiVersion": "1.1.0",  
    "organization": {  
        "id": "CINECA",  
        "name": "CINECA Basel",  
        "description": "Nice description of the CINECA Superstars.",  
        "address": "",  
        "welcomeUrl": "https://www.cineca-project.eu",  
        "contactUrl": "https://twitter.com/cinecaproject",  
        "logoUrl": "https://images.squarespace-cdn.com/content/5cc2c7b2d86cc9589d33dfcc/1556547158578-BXTPXCMXE3BXP283688D/CINECA_log  
o.png?content-type=image%2Fpng",  
        "info": null  
    },  
    "description": "This <a href=\"https://beacon-project.io\">Beacon</a> is based on the GA4GH Beacon <a href=\"https://github.com/  
ga4gh-beacon/specification/blob/v1.1.0/beacon.yaml\">version 1.1.0</a>.",  
    "version": "v1.1.0",  
    "welcomeUrl": "http://localhost:8889",  
    "alternativeUrl": "http://localhost:8889",  
    "createDateTime": "2015-06-01T00:00:00Z",  
    "updateDateTime": "2019-01-07T00:00:00Z",  
    "datasets": [  
        {  
            "id": "1000genomes",  
            "name": null,  
            "description": "Subset of variants of chromosomes 22 and Y from the 1000 genomes project",  
            "assemblyId": "GRCh37",  
            "createDateTime": null,  
            "updateDateTime": null,  
            "dataUseConditions": {  
                "consentCodedataUse": {  
                    "primaryCategory": {  
                        "code": "NRES",  
                        "description": "No restrictions on data use.",  
                        "additionalConstraint": null  
                    },  
                    "secondaryCategories": [],  
                    "requirements": [],  
                    "version": "v1.0"  
                }  
            },  
            "version": null,  
            "variantCount": 3119.  
        }  
    ]  
}
```

Demo

 **CINECA Beacon**
This [Beacon](#) is based on the GA4GH Beacon [version 1.1.0](#)

GRCh37 For example 10 : 1234567890

Included Dataset Responses: All Hit Controlled

Datasets All / None

- 1000genomes (3119 variants)
- urn:hg:example-registered
- urn:hg:example-controlled

Filters + HP:0011007>=49 or PATO:0000383

The EGA archives a large number of datasets, some of which are **Publicly** available. If you have an account, **Controlled** datasets can be done by contacting the relevant Data Access Committee (DAC), whose details are linked to this dataset".

1000genomes	<input checked="" type="checkbox"/>	Subset of variants of chromosomes 22 and Y from the 1000 genomes project
urn:hg:example-registered	<input checked="" type="checkbox"/>	Registered Dataset 1 with fake data (for Bona Fide researchers)
urn:hg:example-controlled	<input checked="" type="checkbox"/>	Controlled Dataset 2 with fake data

Public

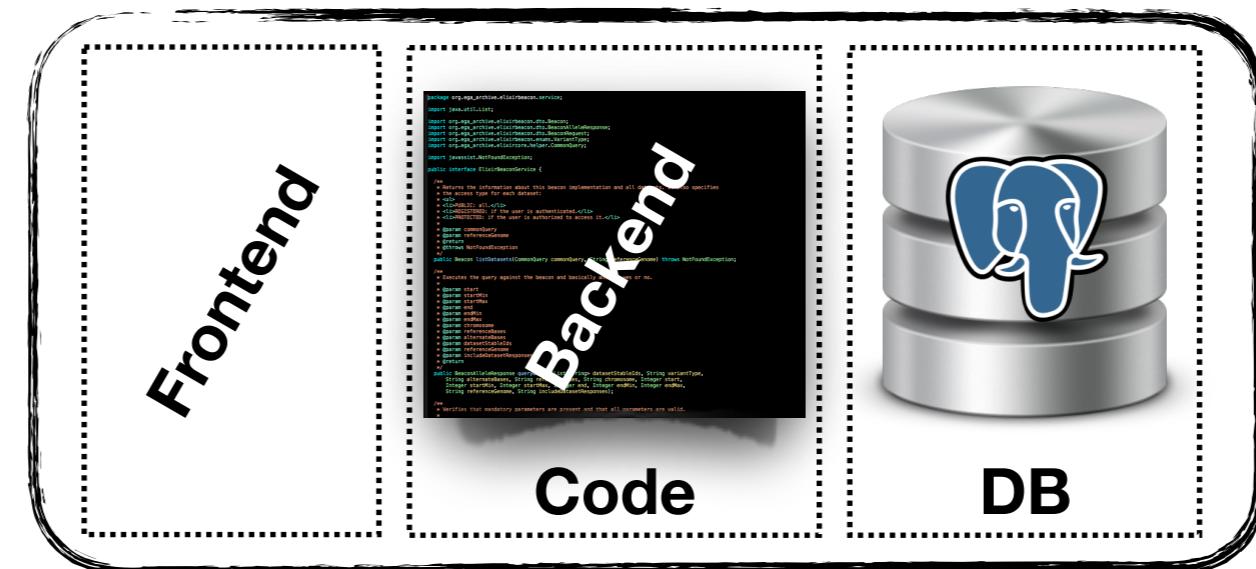
Registered

Controlled

What is it? How to use it?

Beacon N

Demo



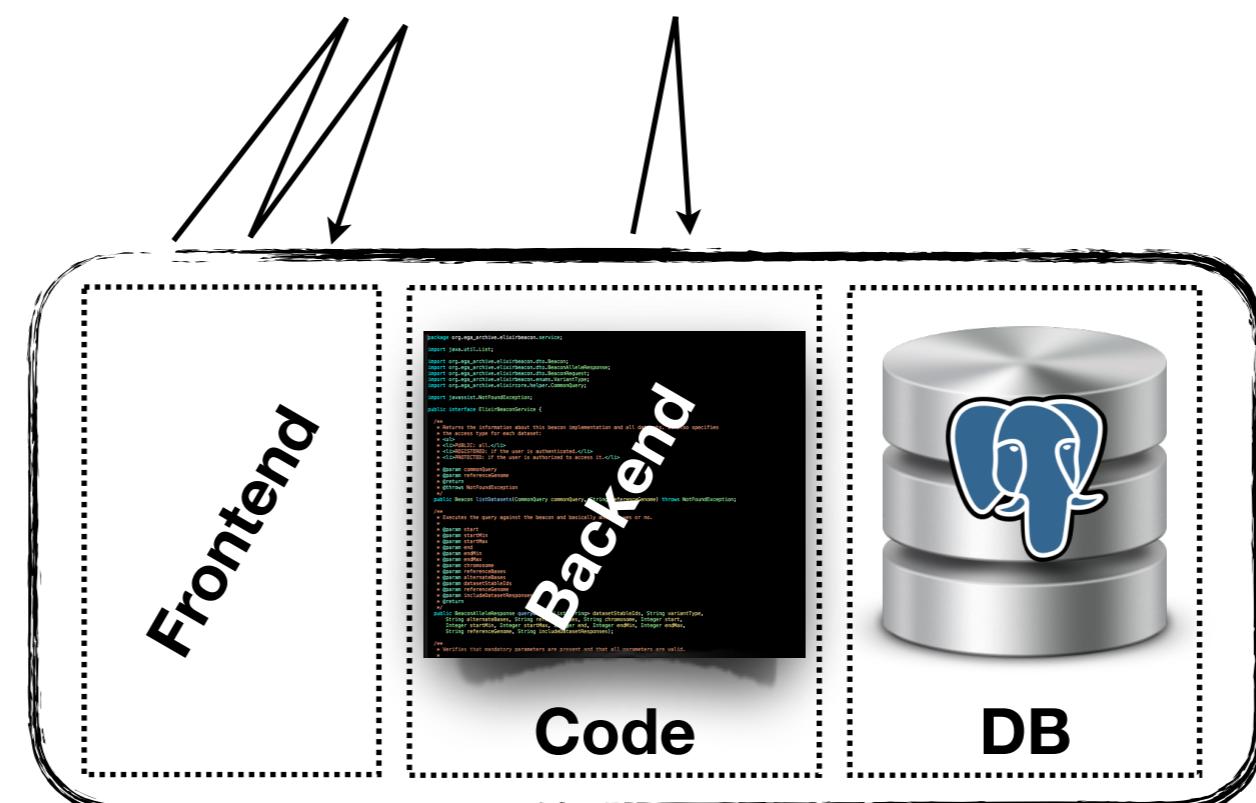
What is it?
How to use it?

Beacon N

Demo



AuthN / AuthZ



What is it?
How to use it?

Demo



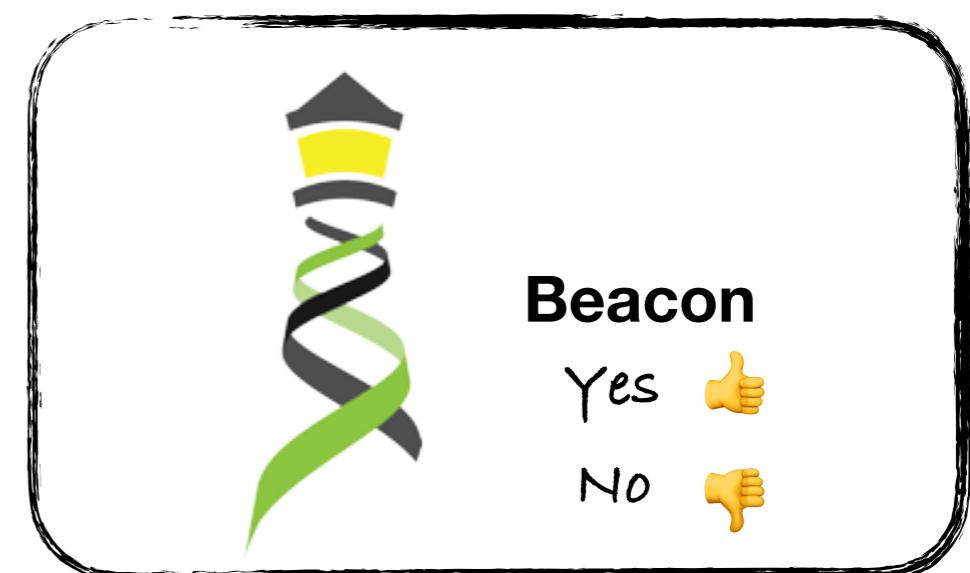
http://beacon/ query ? parameters

- **assemblyId**
- **referenceName**
- **referenceBases**
- **alternateBases**
- **start (startMin / startMax)**
- **end (endMin / endMax)**

- **datasetIds**
- **includeDatasetResponses**
- **variantType**
(eg DUP, DEL, INS, INV, CNV, SNP, MNP)

Specification

1.1.0



including Handovers

What is it?
How to use it?

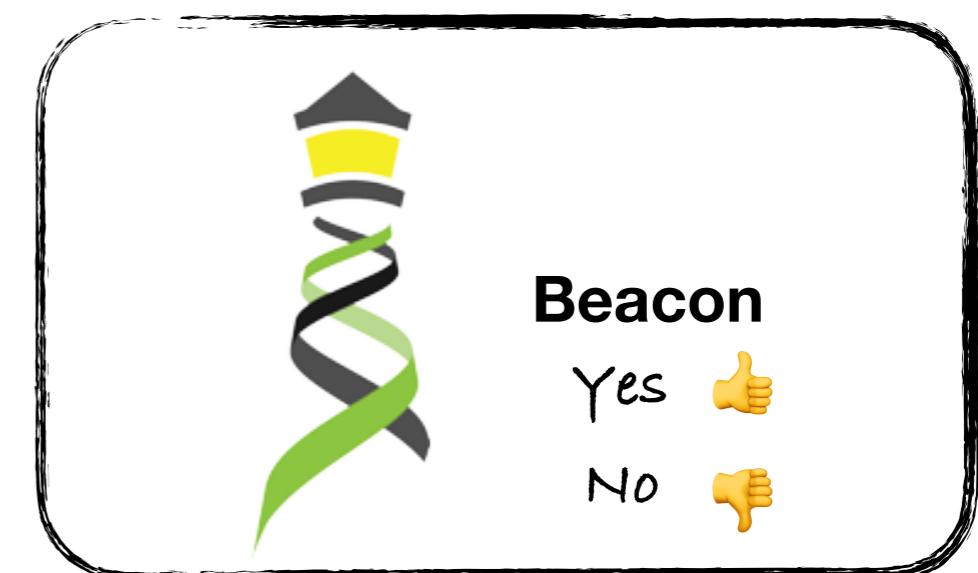
Demo



`http://beacon/ query ? parameters`

- **assemblyId**
- **referenceName**
- **referenceBases**
- **alternateBases**
- **start (startMin / startMax)**
- **end (endMin / endMax)**

Specification



- **datasetIds**
- **includeDatasetResponses**
- **variantType**
(eg DUP, DEL, INS, INV, CNV, SNP, MNP)

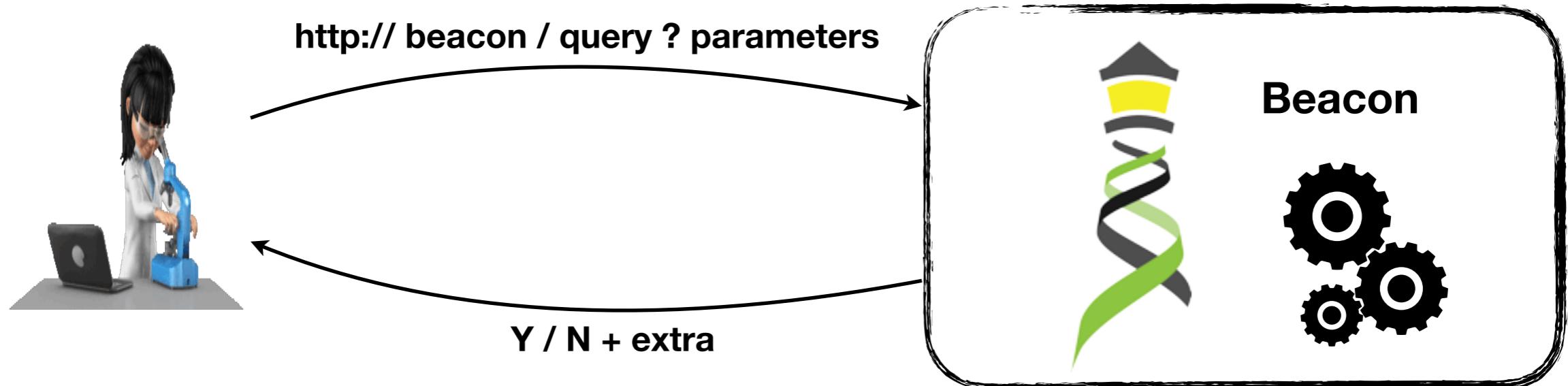
including Handovers

Access Levels 1.5
Filters 2.0

What is it?
How to use it?

Beacon N

Demo



Public

Registered

Controlled

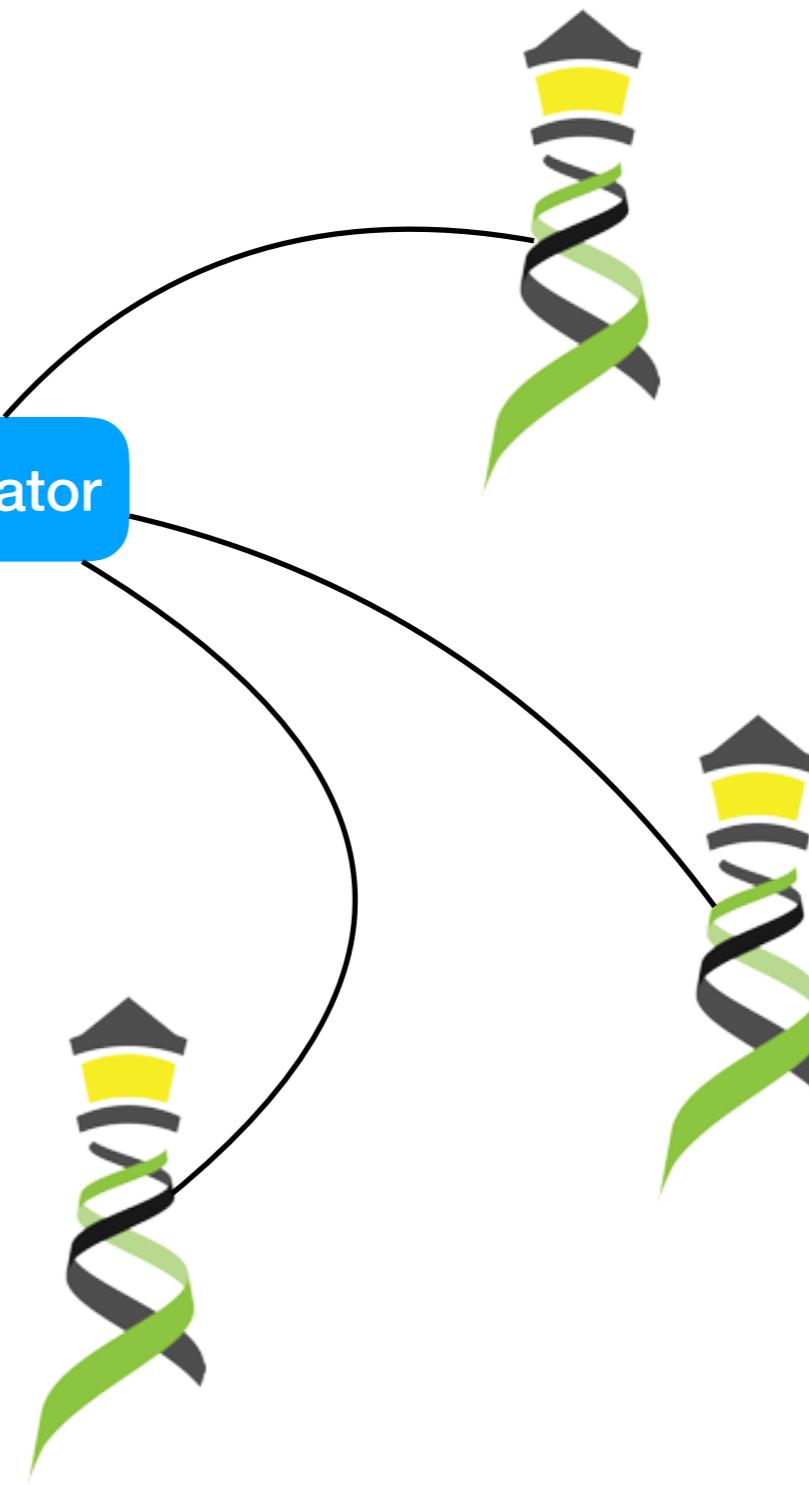
What is it?
How to use it?

Demo

Beacon Network



Aggregator



What is it?
How to use it?

Demo

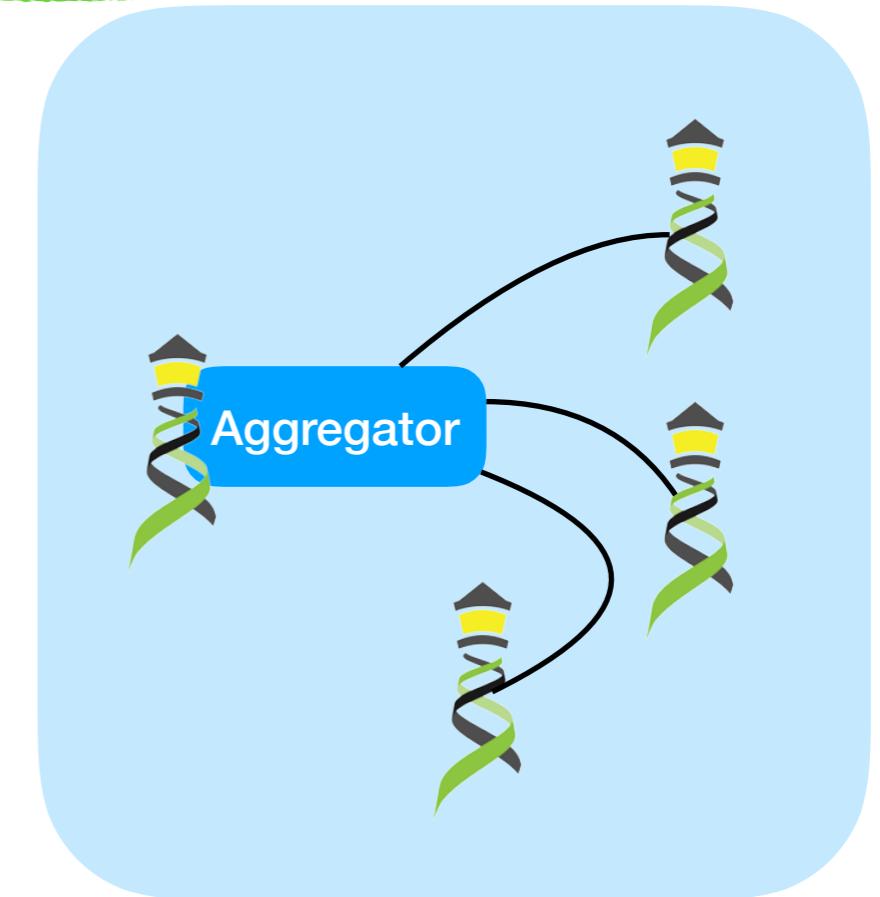
Beacon Network



What is it?
How to use it?

Demo

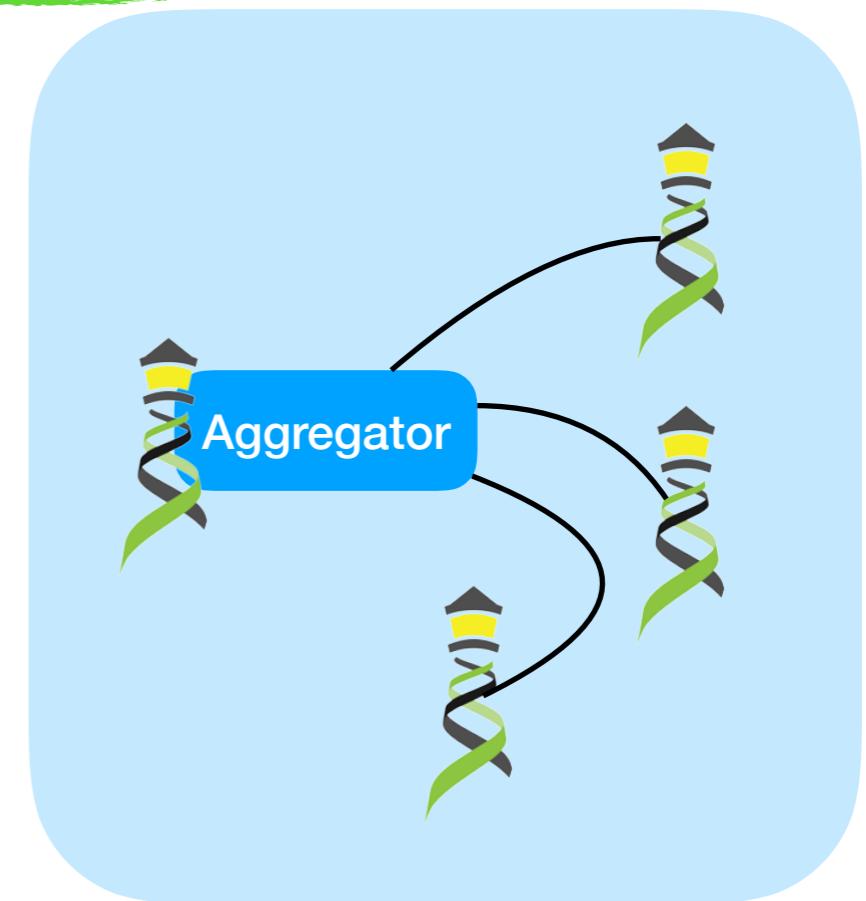
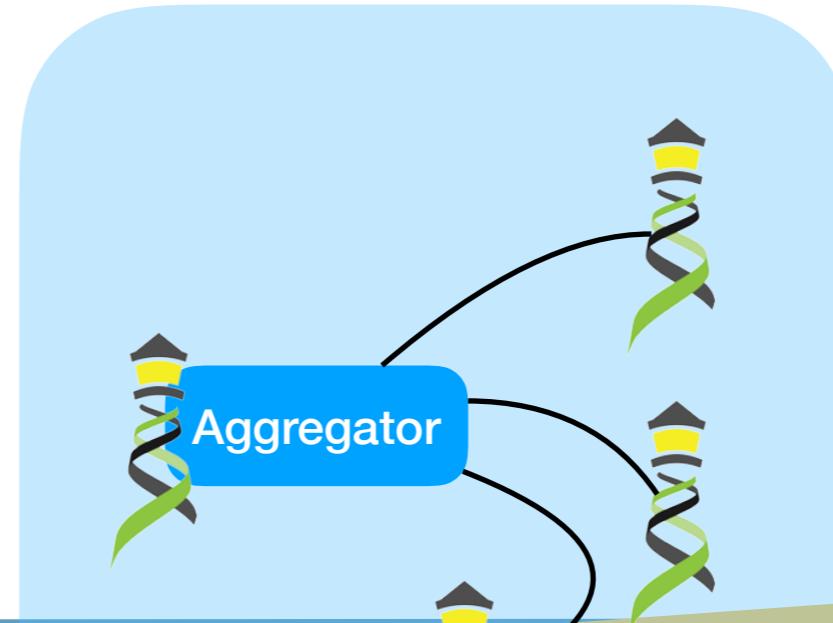
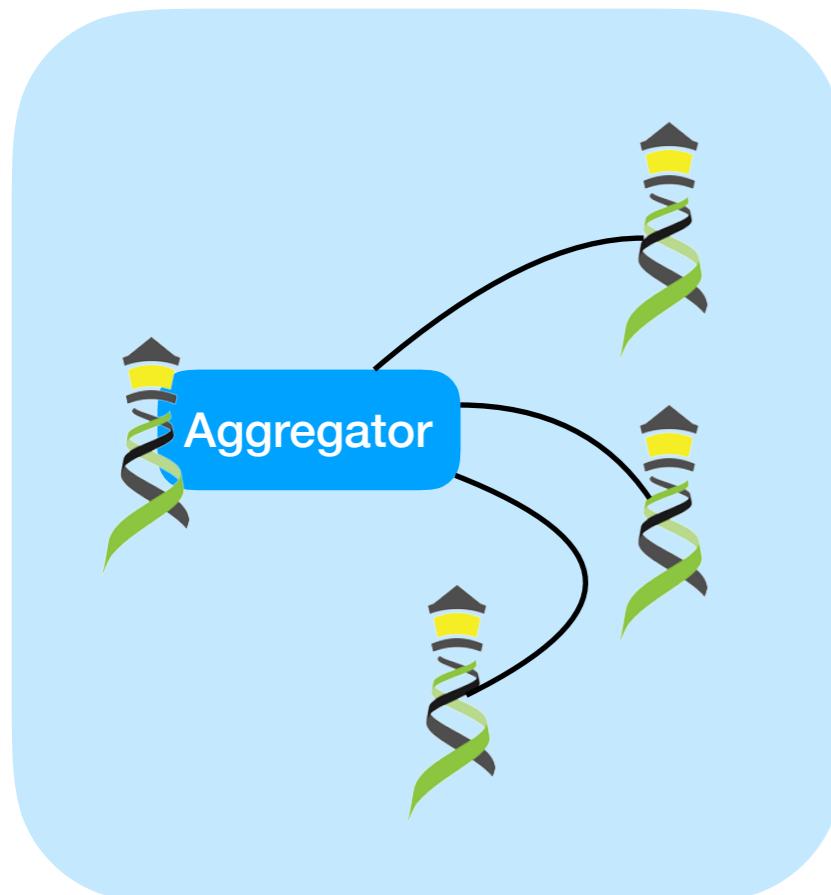
Beacon Network



What is it?
How to use it?

Demo

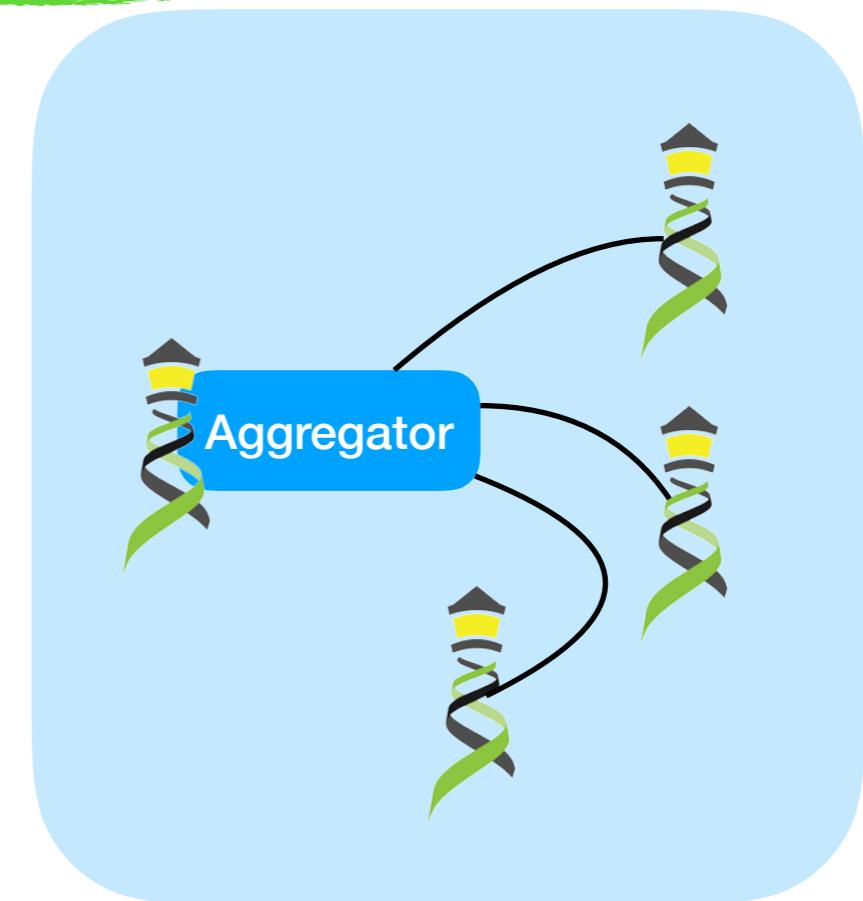
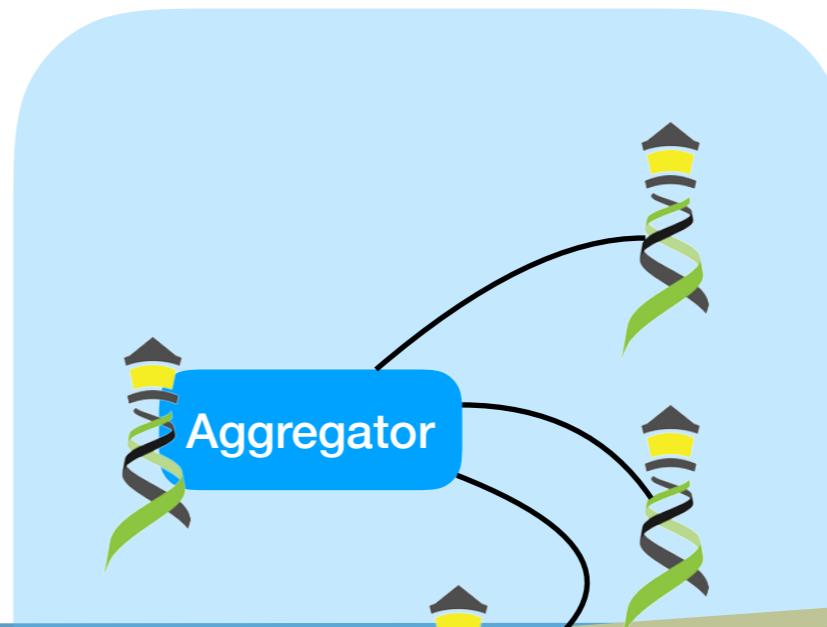
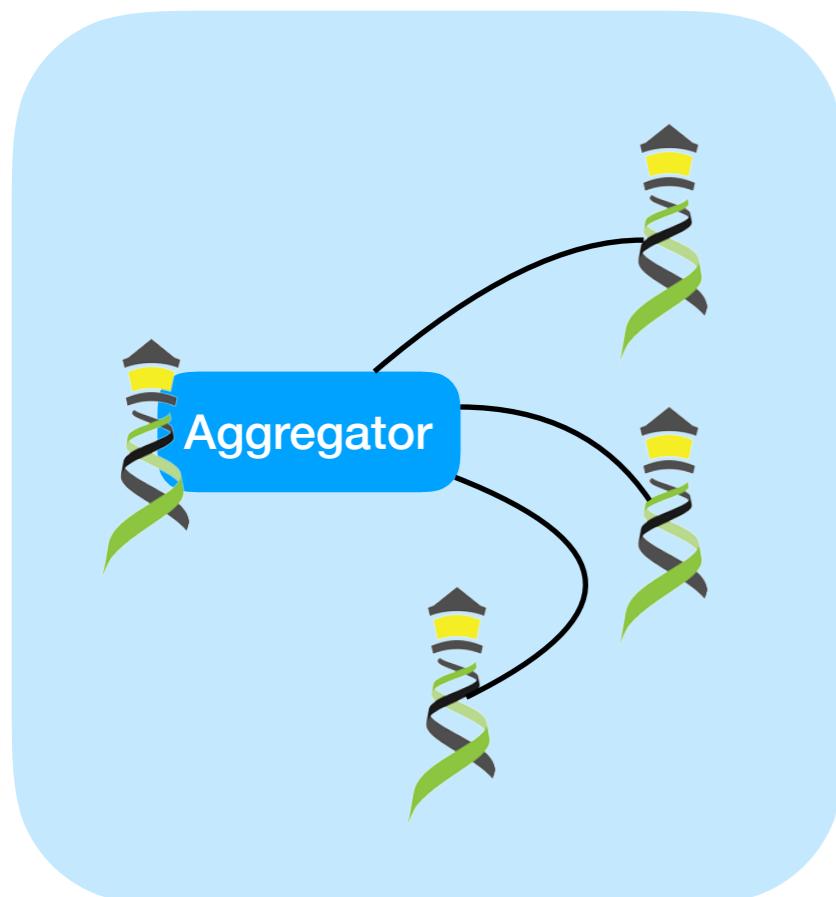
Beacon Network



What is it?
How to use it?

Demo

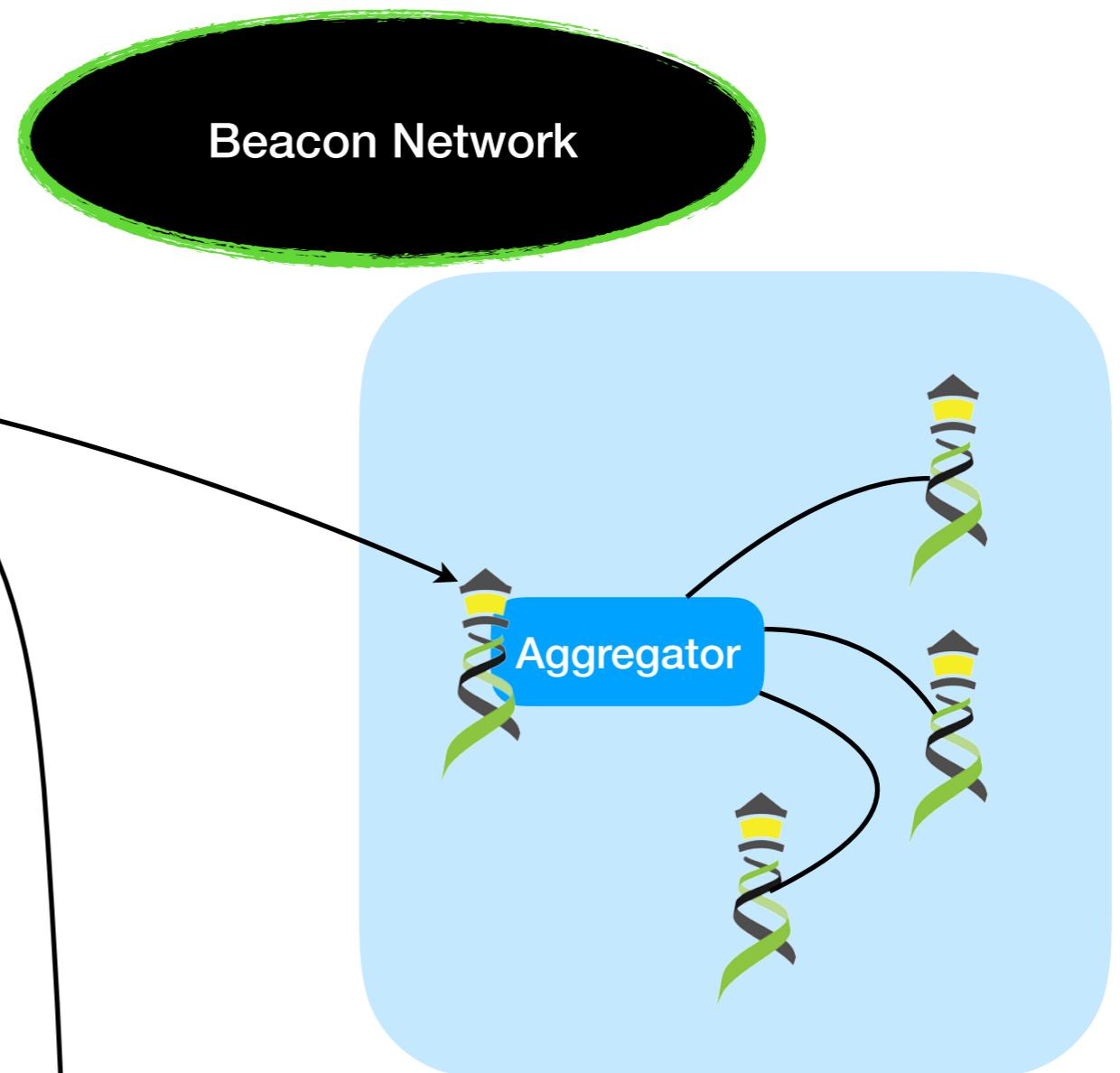
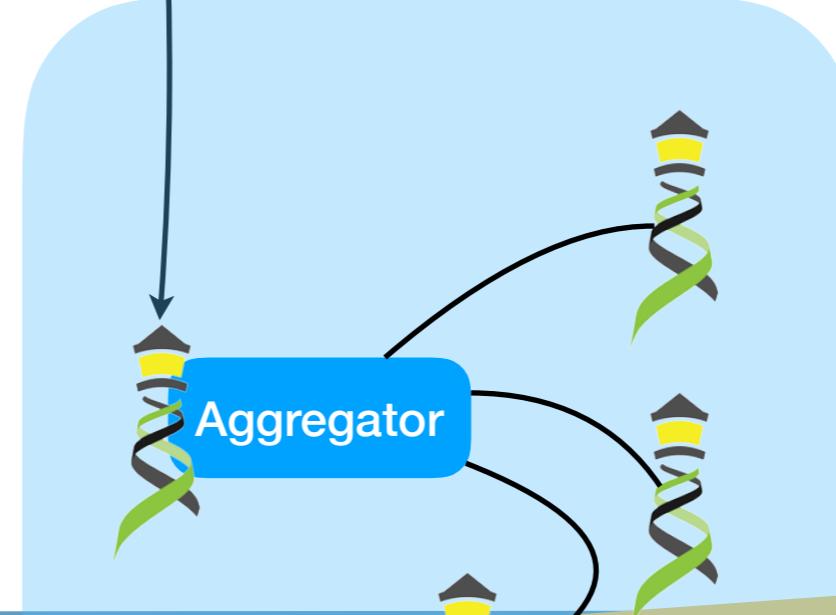
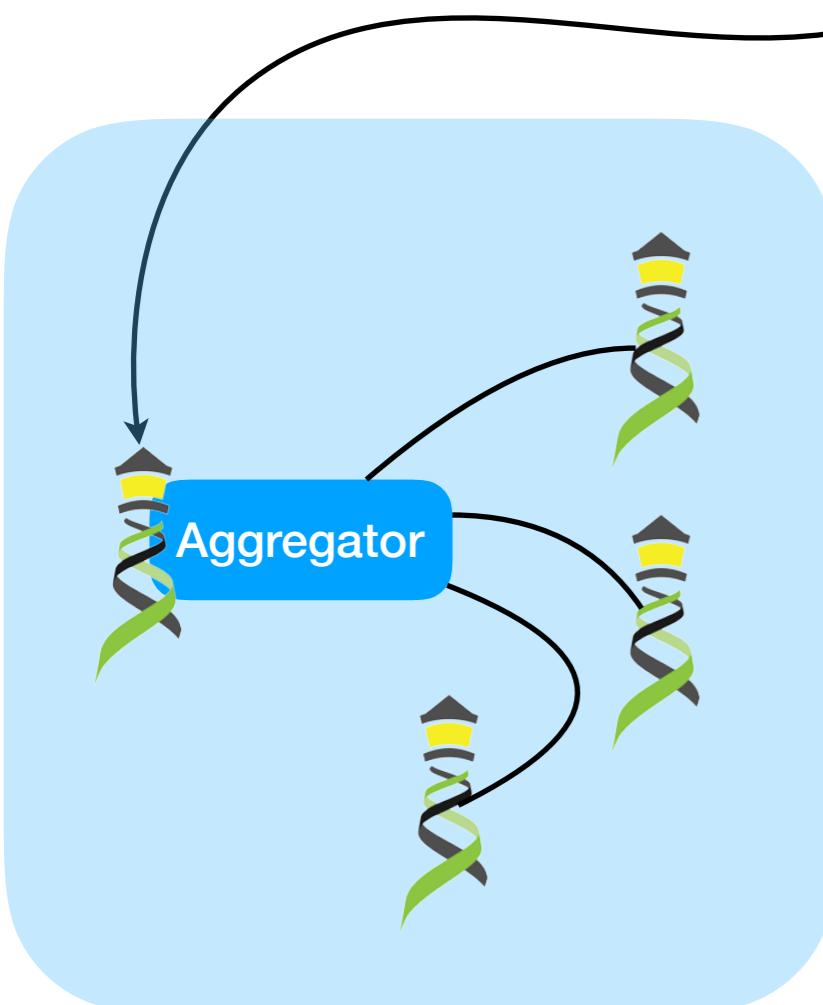
Beacon Network



What is it?
How to use it?

Demo

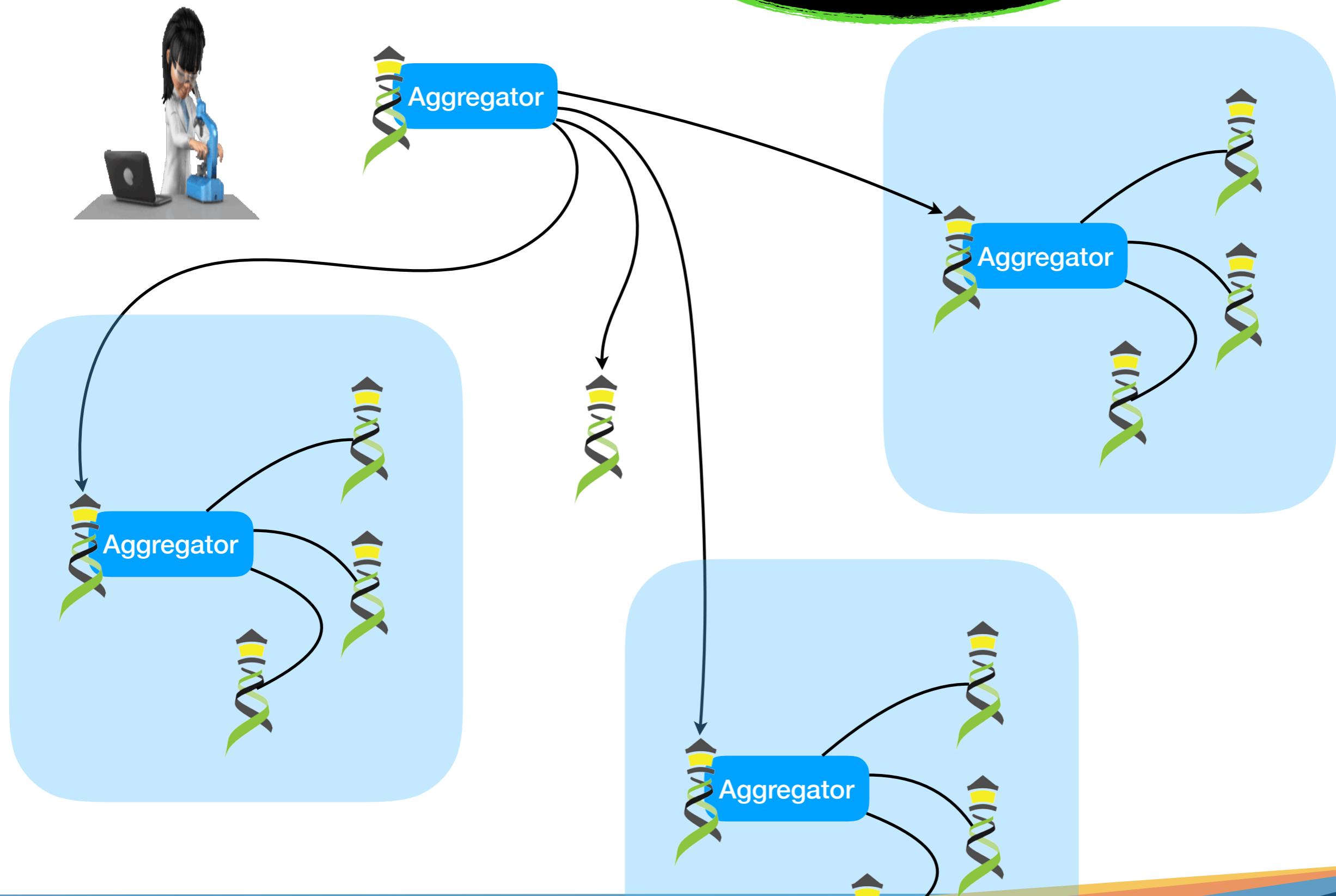
Beacon Network



What is it?
How to use it?

Demo

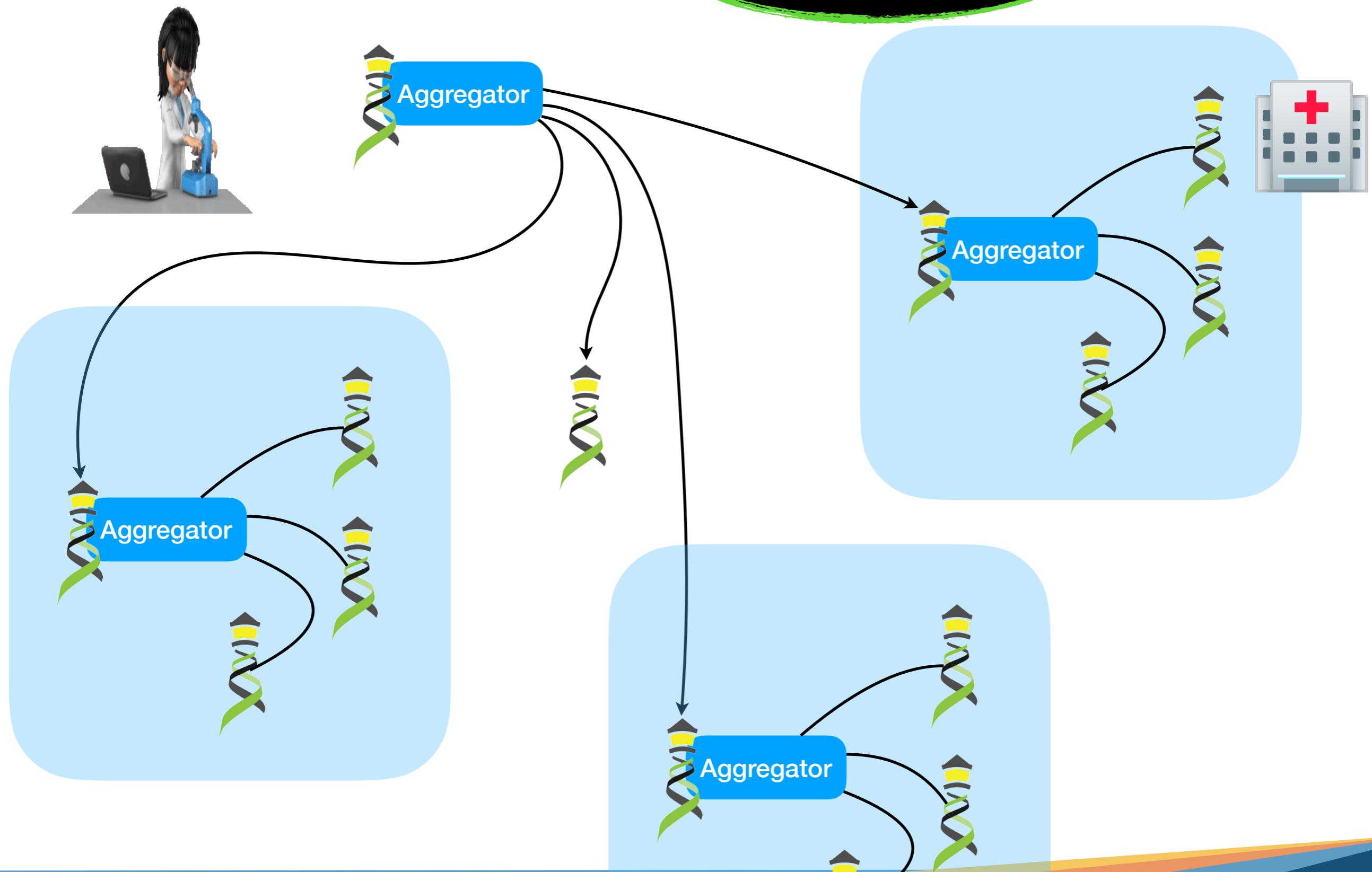
Beacon Network



What is it?
How to use it?

Demo

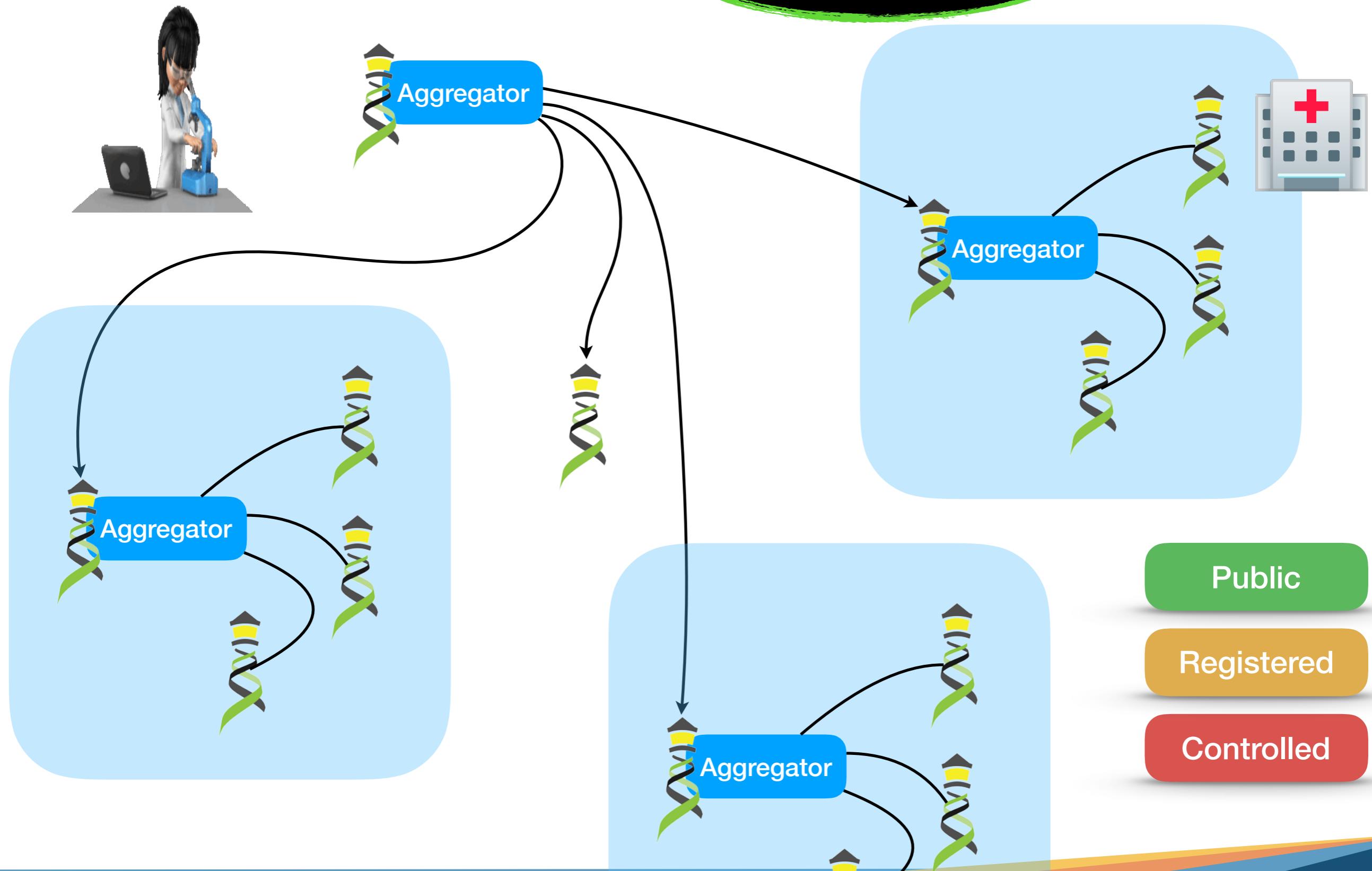
Beacon Network



What is it?
How to use it?

Demo

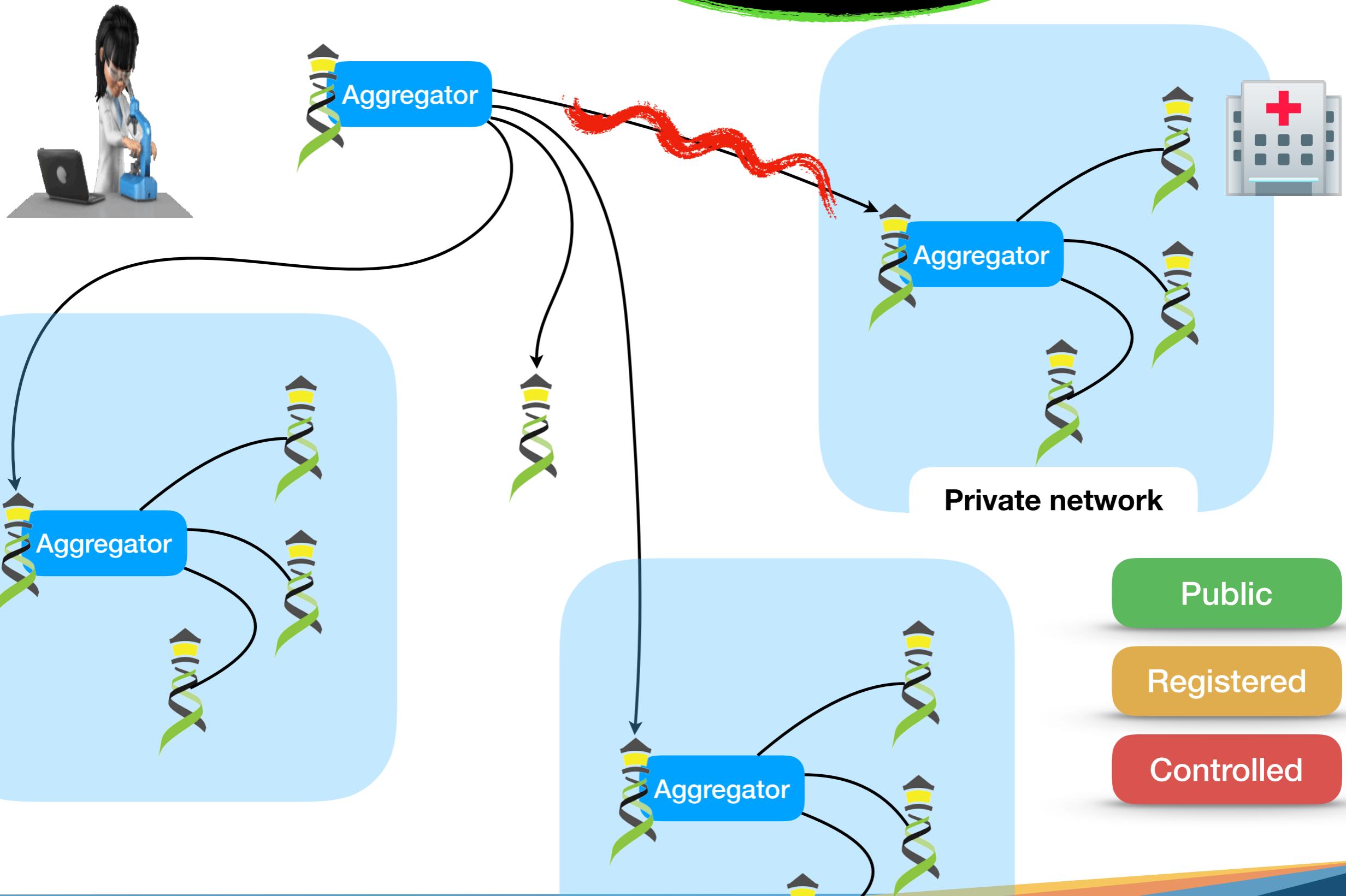
Beacon Network



What is it?
How to use it?

Demo

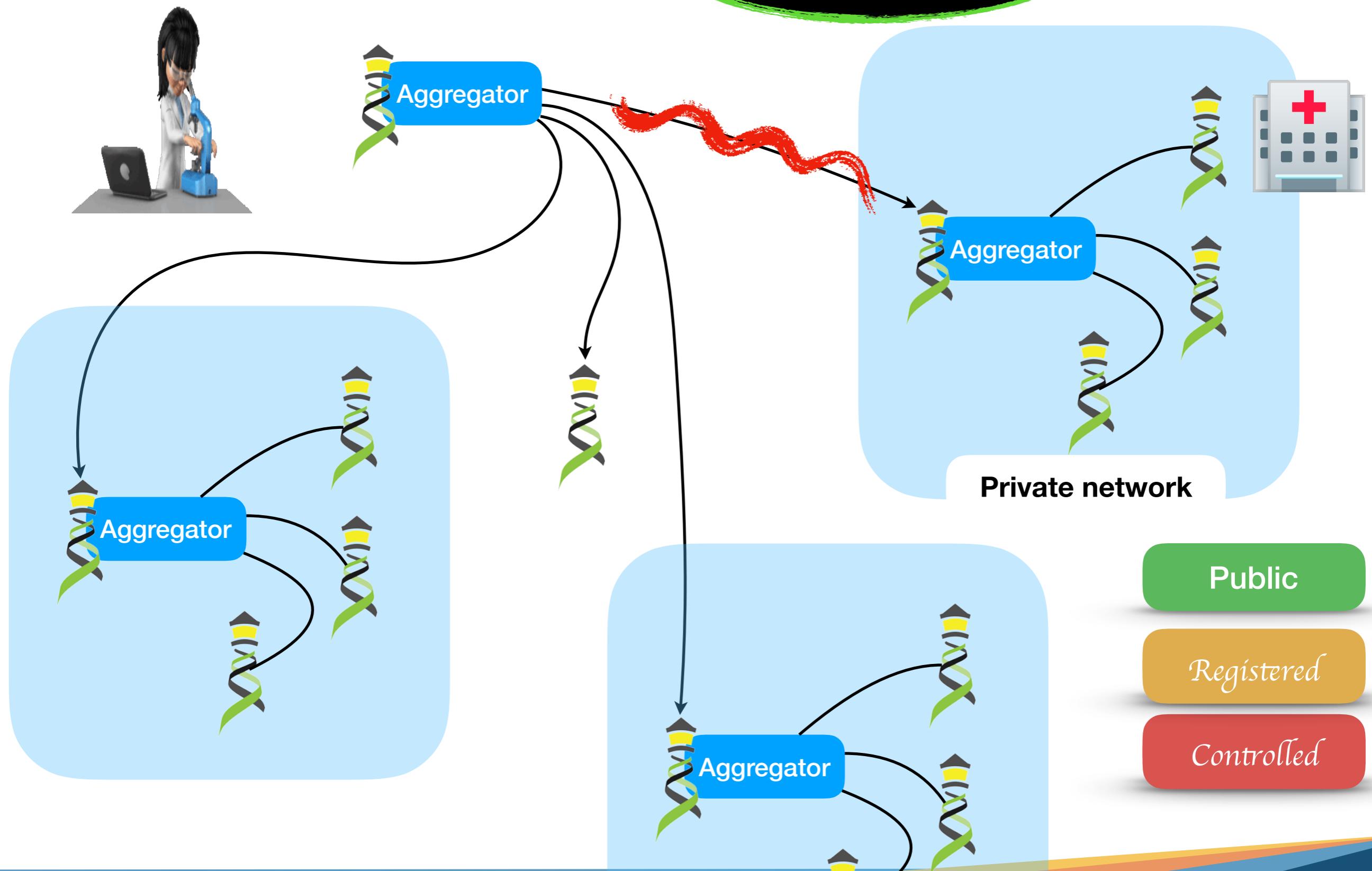
Beacon Network



What is it? How to use it?

Demo

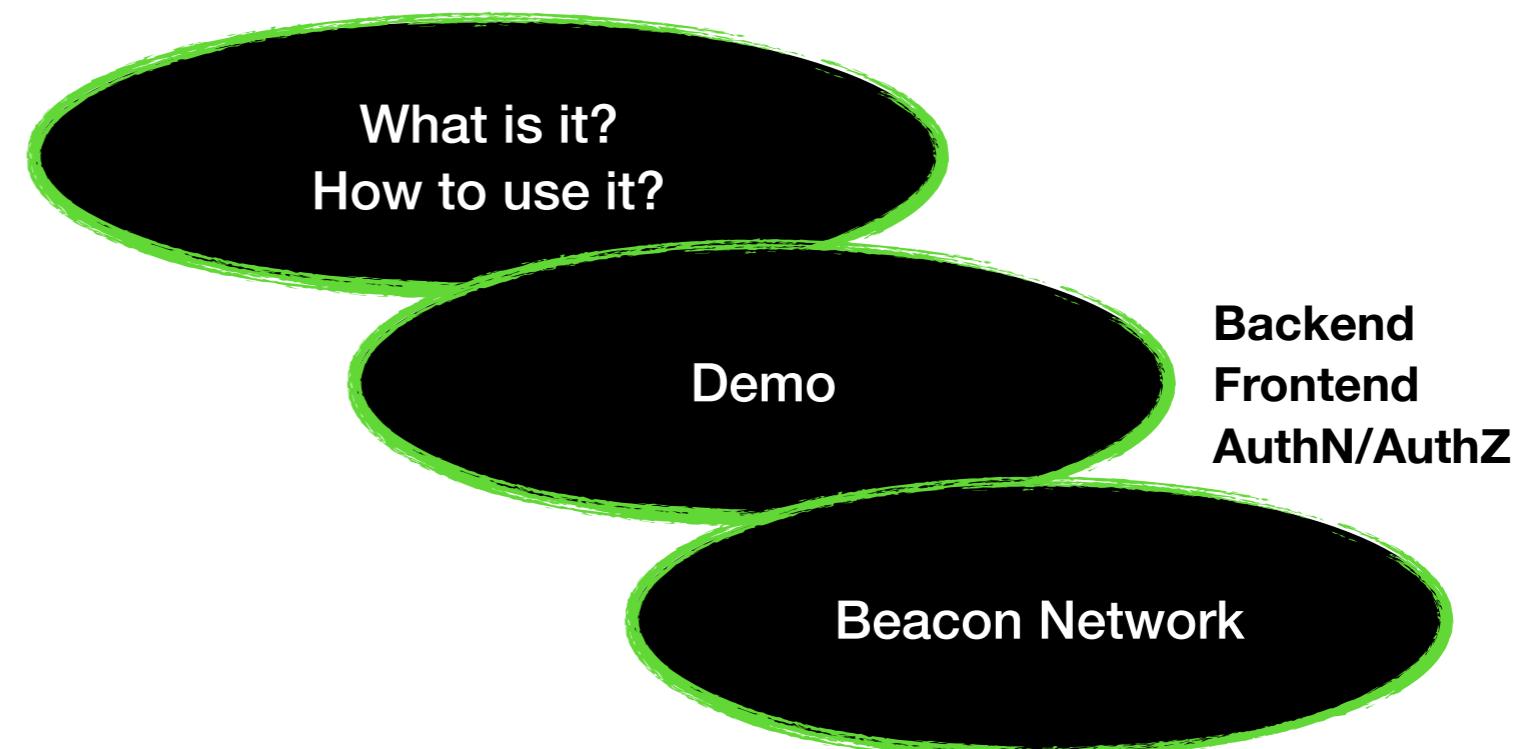
Beacon Network





Beacon

Sabela de la Torre
Frédéric Haziza



Beacon 2.0

On the next major release
of the specification

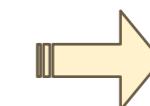
Contents

- One query for each of the existing types
 - SNPs
 - Structural Variants
 - Region
 - ...
- Access levels
- Filters
- New types of queries:
 - By sample
 - By patient
 - ...
- Schema versions



v2.0

Tested and
already
implemented by
real Beacons



GA4GH
approval
process

Ongoing

One query for each type

- beaconGenomicSNPRequest
- beaconGenomicSVRequest (Range or Imprecise)
- beaconGenomicRegionRequest (from Position A to Position B, indirectly: gene, transcript)
- beaconPhenotypeRequest (attributes like “gender” or conditions like “schizophrenia”, Ontology-based queries)
- Also, new queries of a different topic than genomics:
 - beaconProteomic...Request (probably cloning all or similar to the above ones)
 - beaconPlant...Request (probably cloning all or similar to the above ones)

Access Levels

/access_levels

```
{  
  "beaconId": "elixir-demo-beacon",  
  "error": null,  
  "fields": {  
    "beacon": "PUBLIC",  
    "beaconOrganization": "PUBLIC",  
    "beaconDataset": "PUBLIC",  
    "beaconAlleleRequest": "PUBLIC",  
    "beaconGenomicPositionRequest": "REGISTERED",  
    "beaconGenomicRegionRequest": "REGISTERED",  
    "beaconGenomicSVRequest": "NOT_SUPPORTED",  
    "beaconProteomicRequest": "NOT_SUPPORTED",  
    "beaconAlleleResponse": "PUBLIC",  
    "beaconGenomicPositionResponse": "REGISTERED",  
    "beaconGenomicRegionResponse": "REGISTERED"  
    "beaconGenomicSVResponse": "NOT_SUPPORTED",  
    "beaconProteomicResponse": "NOT_SUPPORTED"
```

- NOT_SUPPORTED
- PUBLIC
- REGISTERED
- CONTROLLED

These query types are **not implemented** in this Beacon

Registered access is required to get the **global** response

```
    "beaconAlleleResponses": "PUBLIC",  
    "beaconGenomicPositionResponses": "CONTROLLED",  
    "beaconGenomicRegionResponses": "CONTROLLED",  
    "datasetGenomicSVResponses": "NOT_SUPPORTED",  
    "datasetProteomicResponses": "NOT_SUPPORTED",  
    "beaconError": "PUBLIC",  
    "dataUseConditions": "PUBLIC",  
  },  
  "datasets": {  
    "EGAD000000000001": "PUBLIC",  
    "EGAD000000000002": "CONTROLLED",  
    "EGAD000000000003": "REGISTERED"
```

Controlled access is required to get the **detailed** response

Access Levels - Details by field

/access_levels?includeFieldDetails=true

```
{  
  "beaconId" : "elixir-demo-beacon",  
  "error" : null,  
  "fields" : {  
    "beacon" : {  
      "id" : "PUBLIC",  
      ...  
    },  
    "beaconOrganization": {  
      ...  
    },  
    "beaconDataset": {  
      ...  
    },  
    "beaconAlleleRequest": {  
      ...  
    },  
    "beaconAlleleResponse": {  
      ...  
    },  
    "datasetAlleleResponses": {  
      ...  
      "variantCount": "CONTROLLED",  
      "callCount": "PUBLIC",  
      "sampleCount": "REGISTERED"  
      ...  
    },  
    "beaconError": {  
      ...  
    },  
    "dataUseConditions": {  
      ...  
    },  
    "datasets": {  
      "EGAD00000000003": "PUBLIC",  
      "EGAD00000000002": "CONTROLLED",  
      "EGAD00000000001": "REGISTERED"  
    }  
  }  
}
```

Access level of each field

Summarized info for datasets

Access Levels - Dataset details

```
/access_levels?includeDatasetDetails=true
{
    "beaconId" : "elixir-demo-beacon",
    "error" : null,
    "fields": {
        "beacon": "PUBLIC",
        "beaconOrganization": "PUBLIC",
        "beaconDataset": "PUBLIC",
        "beaconAlleleRequest": "PUBLIC",
        "beaconAlleleResponse": "PUBLIC",
        "datasetAlleleResponses": "PUBLIC",
        "beaconError": "PUBLIC",
        "dataUseConditions": "PUBLIC",
    },
}
```

```
    "datasets" : {
        "EGAD00000000002" : {
            "beaconDataset" : "REGISTERED",
            "datasetAlleleResponses" : "CONTROLLED"
        },
        "EGAD00000000001" : {
            "beaconDataset" : "REGISTERED"
        }
    }
}
```

This should only include those fields which differ from the default value

In this dataset, datasetAlleleResponses has the default value

Filters

```
{  
  "referenceName": "1",  
  "start": 1111,  
  "end": null,  
  "startMin": null,  
  "startMax": null,  
  "endMin": null,  
  "endMax": null,  
  "referenceBases": "A",  
  "alternateBases": "C",  
  "variantType": null,  
  "assemblyId": "GRCh38",  
  "datasetIds": null,  
  "includeDatasetResponses": "ALL",  
  "filters": [ "GO:0019099", "C0006142"]  
}
```

Examples:

- Sex (e.g. Gene Ontology)
- Disease (e.g. ICD10)
- Case/control
- Age
- ...

Filter by: female and
breast cancer

New types of queries: BeaconSampleRequest

```
{  
  "referenceName": "1",  
  "start": 1111,  
  "end": 9999,  
  "startMin": null,  
  "startMax": null,  
  "endMin": null,  
  "endMax": null,  
  "referenceBases": "A",  
  "alternateBases": "C",  
  "variantType": null,  
  "assemblyId": "GRCh38",  
  "datasetIds": null,  
  "includeDatasetResponses": "ALL",  
  "filters": [ "C0006142"]  
}
```

Filter by
breast cancer

Samples corresponding to
patients with disease X
and variants in gene Y

BeaconSampleResponse

```
{  
  "beaconId": "some-beacon",  
  "apiVersion": "2.0",  
  "exists": true,  
  "error": null,  
  "request": {  
    "referenceName": "1",  
    "start": 1111,  
    "end": 9999,  
    "startMin": null,  
    "startMax": null,  
    "endMin": null,  
    "endMax": null,  
    "referenceBases": "A",  
    "alternateBases": "C",  
    "variantType": null,  
    "assemblyId": "GRCh38",  
    "datasetIds": null,  
    "includeDatasetResponses": "ALL",  
    "filters": [ "C0006142"]  
  },  
  ...  
}
```



```
  "samples": [  
    {  
      "age_at_collection" : {...},  
      "biocharacteristics" : [...],  
      "created" : "2017-10-25T07:06:03Z",  
      "data_use_conditions" : {...},  
      "description" : "Burkitt lymphoma",  
      "external_references" : [...],  
      "geo_provenance" : {...},  
      "id" : "AM_BS_NCBISKYCGH-1993",  
      "individual_id" : "ind-cnhl-1293347-004",  
      "info" : {...},  
      "name" : "Sample BRCA-00429, 2nd replicate",  
      "project_id" : "ind-cnhl-1293347-004",  
      "updated" : "2017-10-25T07:06:03Z"  
    },  
    ...  
  ]  
}
```

<https://schemablocks.org/schemas/blocks/Biosample.html>

New types of queries: BeaconPatientRequest

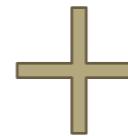
```
{  
  "referenceName": "1",  
  "start": 1111,  
  "end": 9999,  
  "startMin": null,  
  "startMax": null,  
  "endMin": null,  
  "endMax": null,  
  "referenceBases": "A",  
  "alternateBases": "C",  
  "variantType": null,  
  "assemblyId": "GRCh38",  
  "datasetIds": null,  
  "includeDatasetResponses": "ALL",  
  "filters": [ "C0006142", "BTO:0000199"]  
}
```

Patients with disease X
and variants in gene Y
that have samples of
tissue Z available

Filter by breast
cancer & cardiac
muscle

BeaconPatientResponse

```
{  
  "beaconId": "some-beacon",  
  "apiVersion": "2.0",  
  "exists": true,  
  "error": null,  
  "request": {  
    "referenceName": "1",  
    "start": 1111,  
    "end": 9999,  
    "startMin": null,  
    "startMax": null,  
    "endMin": null,  
    "endMax": null,  
    "referenceBases": "A",  
    "alternateBases": "C",  
    "variantType": null,  
    "assemblyId": "GRCh38",  
    "datasetIds": null,  
    "includeDatasetResponses": "ALL",  
    "filters": [ "C0006142", "BTO:0000199"]  
  },  
  ...  
}
```



[https://schemablocks.org/
schemas/blocks/Individual.html](https://schemablocks.org/schemas/blocks/Individual.html)

```
"patients": [  
  {  
    "biocharacteristics" : [...],  
    "created" : "2017-10-25T07:06:03Z",  
    "data_use_conditions" : {...},  
    "description" : "patient with lung cancer, male smoker",  
    "external_references" : [...],  
    "geo_provenance" : {...},  
    "id" : "AM_BS__NCBISKYCGH-1993",  
    "info" : {...},  
    "name" : "Ion Tichy, space explorer",  
    "organism" : {  
      "id" : "NCBITaxon:9606",  
      "label" : "Homo sapiens"  
    },  
    "sex" : {  
      "id" : "PATO:0020000",  
      "label" : "female genetic sex"  
    },  
    "updated" : "2017-10-25T07:06:03Z"  
  },  
  ...  
]
```

Schema versions

- Beacons can declare which versions they support and which is the default one.
- Users can choose among these versions to get a response compliant with that version.

Versions of other schemas referenced in the Beacon (e.g. Biosample in Schema Blocks)

```
{  
  "meta": {  
    "request": {  
      "samples": "1.2"  
    },  
    "apiVersion": "1.1.0"  
  },  
  "query": {  
    "referenceBases": null,  
    "referenceName": "10",  
    "start": 100006353,  
    "end": 100006356,  
    "assemblyId": "GRCh37",  
    "datasetIds": null,  
    "includeDatasetResponses":  
      "HIT",  
    "filters": null  
  }  
}
```

Beacon 2.0

On the next major release
of the specification