

Accessing data the right way!

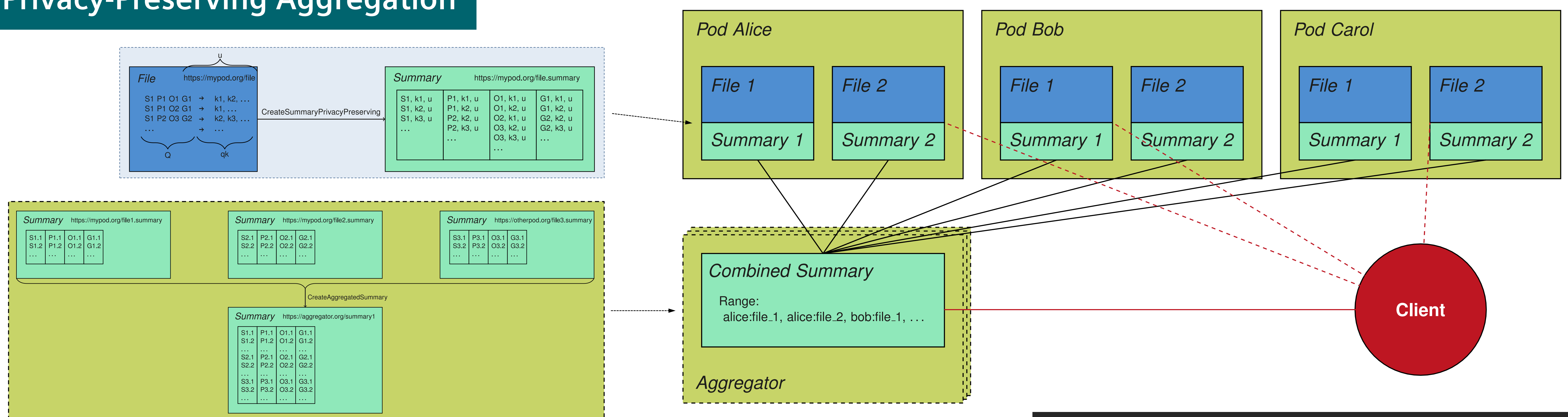
Privacy-preserving querying over decentralized environments

In **decentralized environments** such as Solid **not all data is necessarily public**

Query engines need to be able to **authenticate** themselves to sources

Privacy-preserving aggregation techniques need to be introduced, so that **untrusted third-parties** are not able to read data

Privacy-Preserving Aggregation



Fine-grained Access control

```
<http://alice.pod/share/policy> a odr:Policy ;
odrl:permission [
  odr:target [
    a odr:AssetCollection ;
    odr:source <https://alice.databox.me/docs/file1> ;
    odr:refinement [ # refine the asset/resource
      a sh:NodeShape ;
      sh:pattern "public" ; sh:flags "i"
    ] ;
    odr:action acl:Read ; ] .
```

Shape the resource that can be accessed

```
<http://alice.pod/share/policy> a odr:Policy ;
odrl:permission [
  odr:target <http://alice.pod/share/photoAnnotations/> ;
  odr:action acl:Read ;
  odr:assignee [
    a foaf:Agent ;
    odr:source <https://alice.example.com/work-groups#Accounting>;
    odr:refinement [ # refine the agent
      a sh:NodeShape ;
      sh:property [
        sh:path foaf:interest ; sh:minCount 1 ;
        sh:hasValue <#photography> ]]]] .
```

Shape the agents/party the policy applies to

```
<http://alice.pod/share/policy> a odr:Policy ;
odrl:permission [
  odr:target <http://alice.pod/share/photoAnnotations/> ;
  odr:action [
    rdf:value acl:Read ;
    odr:refinement [# refine the agent
      a sh:NodeShape ;
      sh:targetObjectsOf ldp:contains ;
      sh:rule [ a sh:SPARQLRule ;
        sh:construct "" CONSTRUCT WHERE { $this a ?o . } "" ;
        sh:condition [
          sh:pattern "private" ; sh:flags "i" ;
          sh:property [
            sh:path [ sh:inversePath ldp:contains ] ;
            sh:hasValue <http://alice.pod/share/photoAnnotations/> ;
            sh:minCount 1 . ]]]]]] .
```

Shape the action/mode that can be performed on the data

Privacy-Preserving Federated Querying

