Hyperledger Fabric

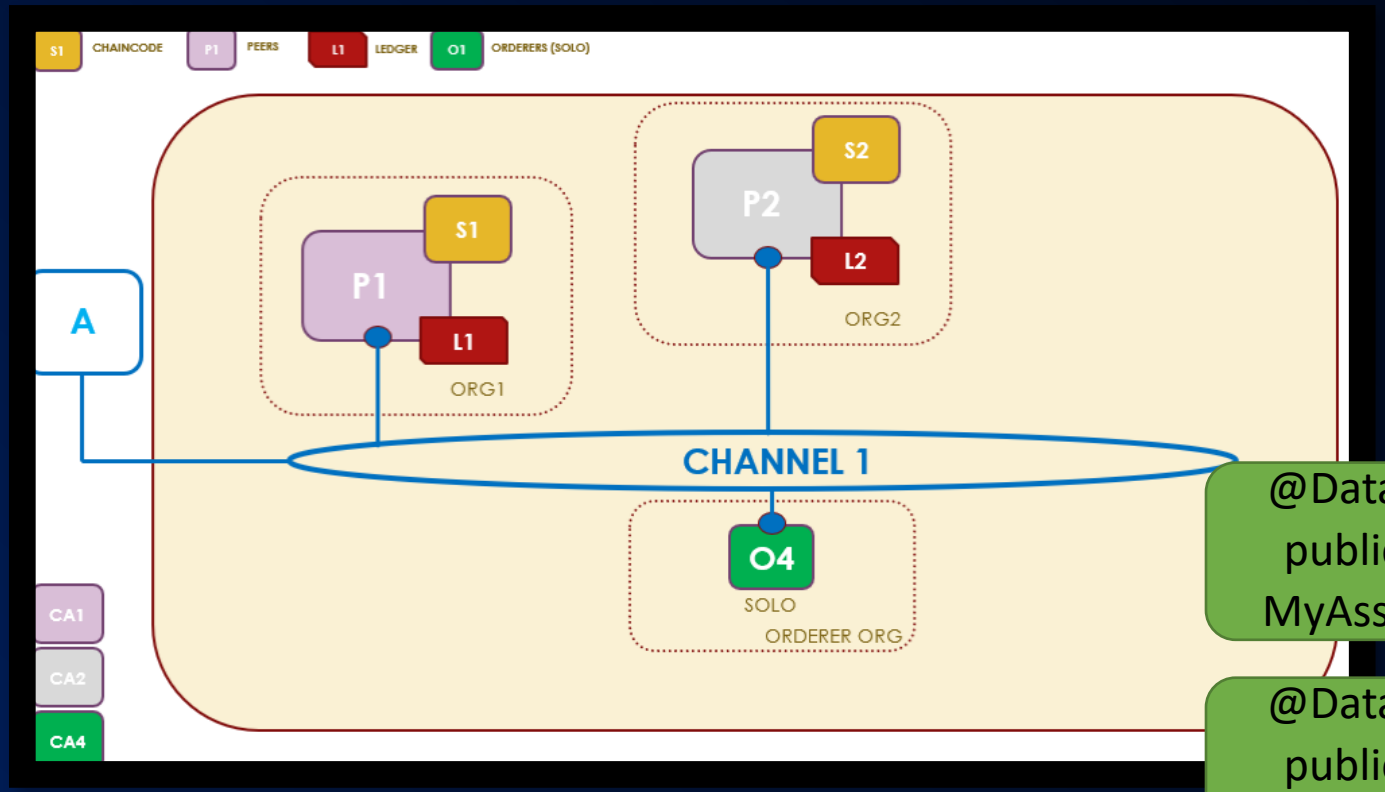
Final exercise

Stefano Avola,
Email: s.avola@cipi.unige.it or s.avola@doc-space.net

Outline

- HLF review
- HLF Exercise
  - Smart Contract
  - Client Application
- What's Next?

HLF review




```
@DataType()
public class
MyAsset { … }
```

```
@DataType()
public class
MyAsset2 { … }
```

```
@DataType()
public class
MyAsset3 { … }
```

```
MySmartContract … {

    @Transaction(…)
    public MyAsset transact1() {
      …
    }

    @Transaction(…)
    public MyAsset3 transact2() {
      …
    }
}
```

Ledger

HLF exercise

- «**Pittaluga & fratelli**» is a big multinationals with headquarters in Genoa and it is specialized in the basil transport sector. Each basil plant is traced at every point: from the greenhouse to the neighborhood supermarket. The specialty of «Pittaluga & fratelli» is to provide a service which, by scanning a special QR placed on the plant pot, provides the entire trace of the basil and for each single leg of the transport chain it returns its owner, together with other general information (temperature, humidity, date, time, …)
- There is one (or more than one) application allowing you to trace in real time the path of each plant currently in possession.
  - (You could develop an app switching between an org and another!)
- Develop the system using Permissioned Blockchain (Hyperledger Fabric).

- (any reference is purely coincidental)

HLF exercise - Notes

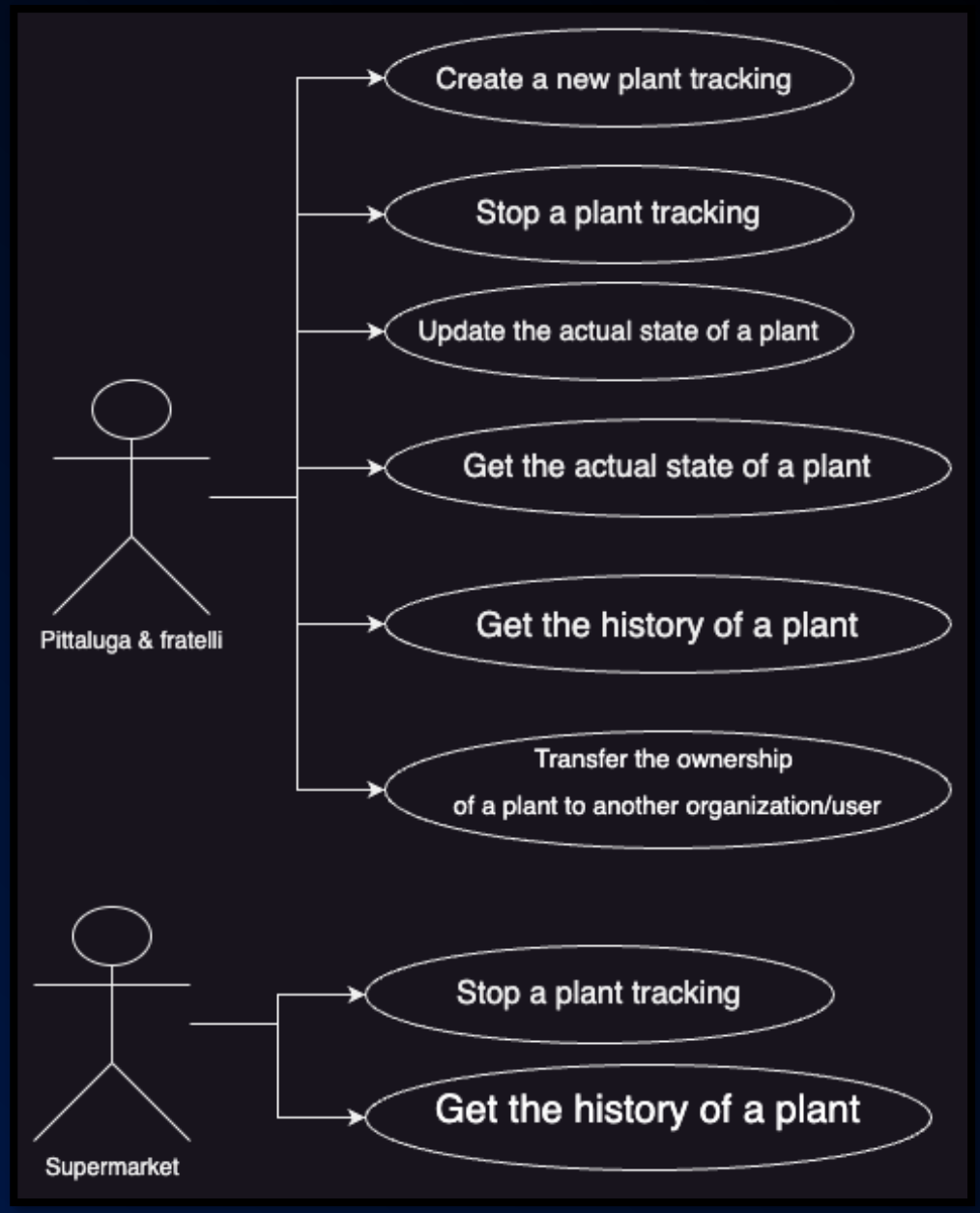
- Org1 = «Pittaluga & fratelli»
- Org2 = Supermarket
- QR code = made-up string (ID) that uniquely defines a plant

HLF exercise – Interviews and Requirements

- Let’s consider those as already done and let's move on to the use cases
- Choice:
  - application should be able to connect to both organizations, or
  - develop an application for each org

HLF exercise – Use Cases

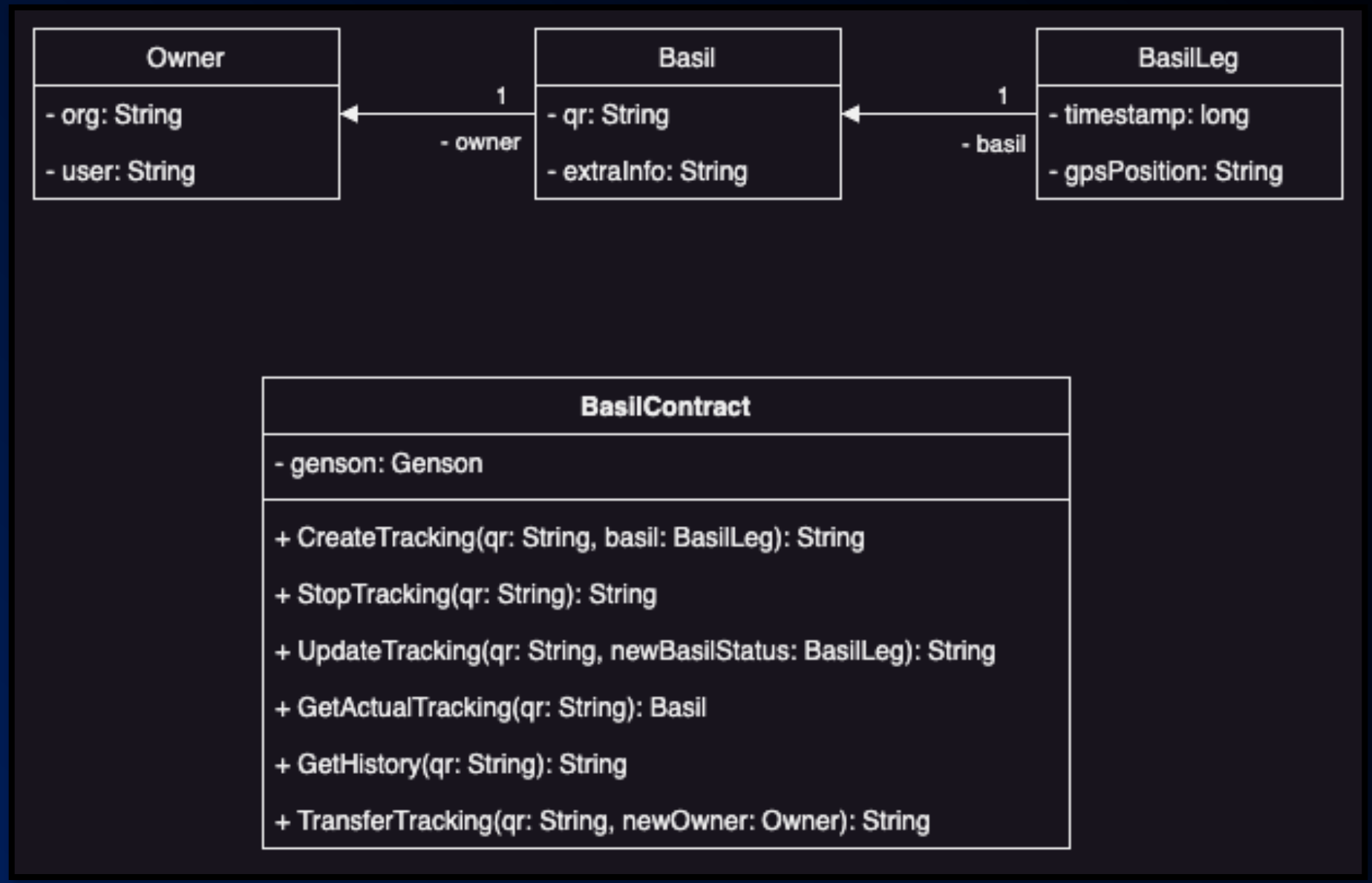
- Smart Contract & Application(s) allow executing the same operations



HLF exercise – Use Cases

Create a new plant tracking	• Given a QR which does not exist yet, create a plant having as its owner the organization that submitted the transaction + creation timestamp and null gps
Stop a plant tracking	• Given an existing QR, delete the plant, provided that who submitted the transaction is the owner organization of the plant.
Update the actual state of a plant	• Given an existing QR, update the state of the associated plant with its new passed value + gps and timestamp, provided that who submitted the transaction is the owner organization of the plant.
Get the actual state of a plant	• Given a QR, return the actual state of the associated plant without checking permissions.
Get the history of a plant	• Given a QR, return the history of the associated plant without checking permissions.
Transfer the ownership of a plant to another organization	• Given an existing QR, an organization, change the owner organization of the plant with the ones passed, provided that who submitted the transaction is the owner of the plant. • Note: if the organization is the same as the actual one of the plant, overwriting it is not a problem
PAY ATTENTION: the supermarket cannot execute all the operations!	

HLF exercise – Smart Contract - Class Diagrams



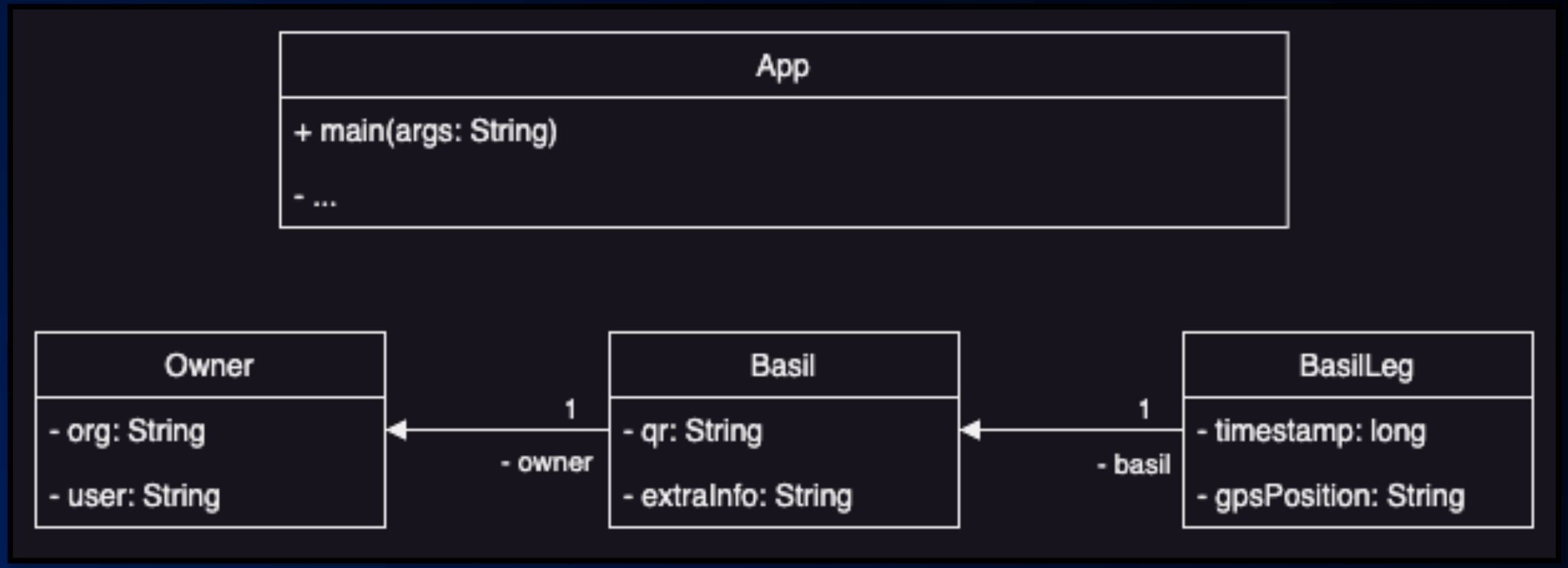
- Implement the default constructor of `Basil`, `BasilLeg` and `Owner`!

HLF exercise – Developing the Smart Contract

Steps:

1. Create the network if not already done (or restart it)
2. Import the chaincode-template project
   - pay attention to the (gradle ...) project
3. Implement `Owner`, `Basil` and `BasilLeg` DataType classes
4. Implement `BasilContract` with its methods/transactions
5. Deploy the chaincode
6. For next chaincode deployments remember to use the `-ccv` flag

HLF exercise – Client Application - Class Diagrams



HLF exercise – Developing client application

Steps:

1. Import the application-template project
   - pay attention to the project name and to the PATH_TO_TEST_NETWORK field!!
2. Implement `Owner`, `Basil` and `BasilLeg` as usual classes
   - It would be better to create an ad hoc library/jar
3. Choose whether:
   - To implement the application so that you can switch between different organizations, or
   - To create 1 App for each organization
4. Implement the App that:
   1. Connects to the channel,
   2. Provides an interactive menu or a GUI,
   3. Invokes the chaincode and prints results.
5. Run the App on the same machine where the channel is.

What's next?

- Do the exercise and show it working: it is a minimum requirement to access the oral exam.

- Who shows that it is works properly to me by the end of January will not have to present it on the day of the oral exam.
  - From February onwards you will have to show it directly to the professor on the day of the oral exam.

- Good luck!