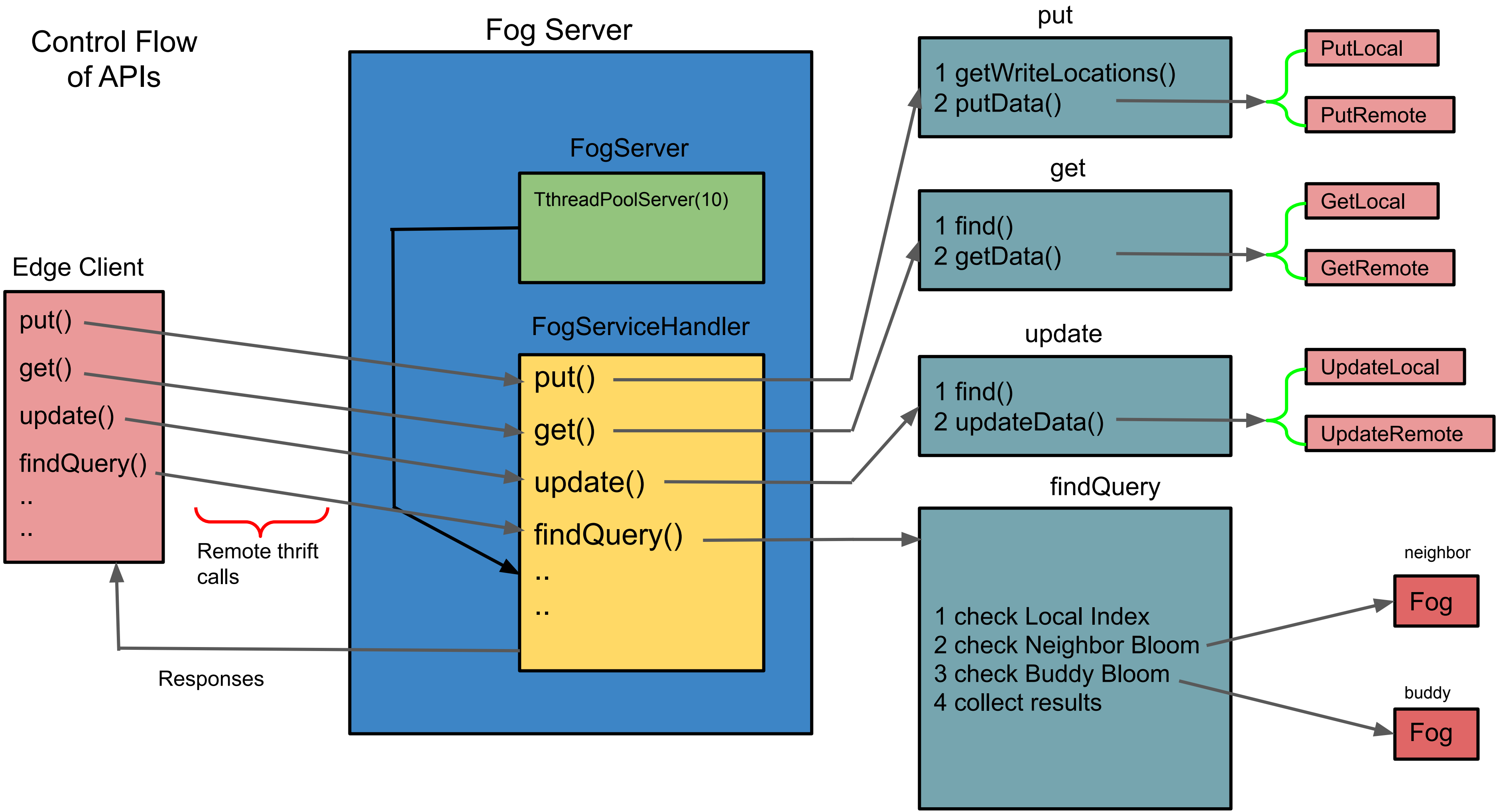
Elfstore Architecture

Sheshadri K R

Control Flow of APIs



Fog Server

FogServer

TthreadPoolServer(10)

FogServiceHandler

put()

get()

update()

findQuery()

..

..

Edge Client

put()

get()

update()

findQuery()

..

..

Remote thrift calls

Responses

put

1 getWriteLocations()
2 putData()

PutLocal

PutRemote

get

1 find()
2 getData()

GetLocal

GetRemote

update

1 find()
2 updateData()

UpdateLocal

UpdateRemote

findQuery

1 check Local Index
2 check Neighbor Bloom
3 check Buddy Bloom
4 collect results

neighbor

Fog

buddy

Fog

Elfstore - Client

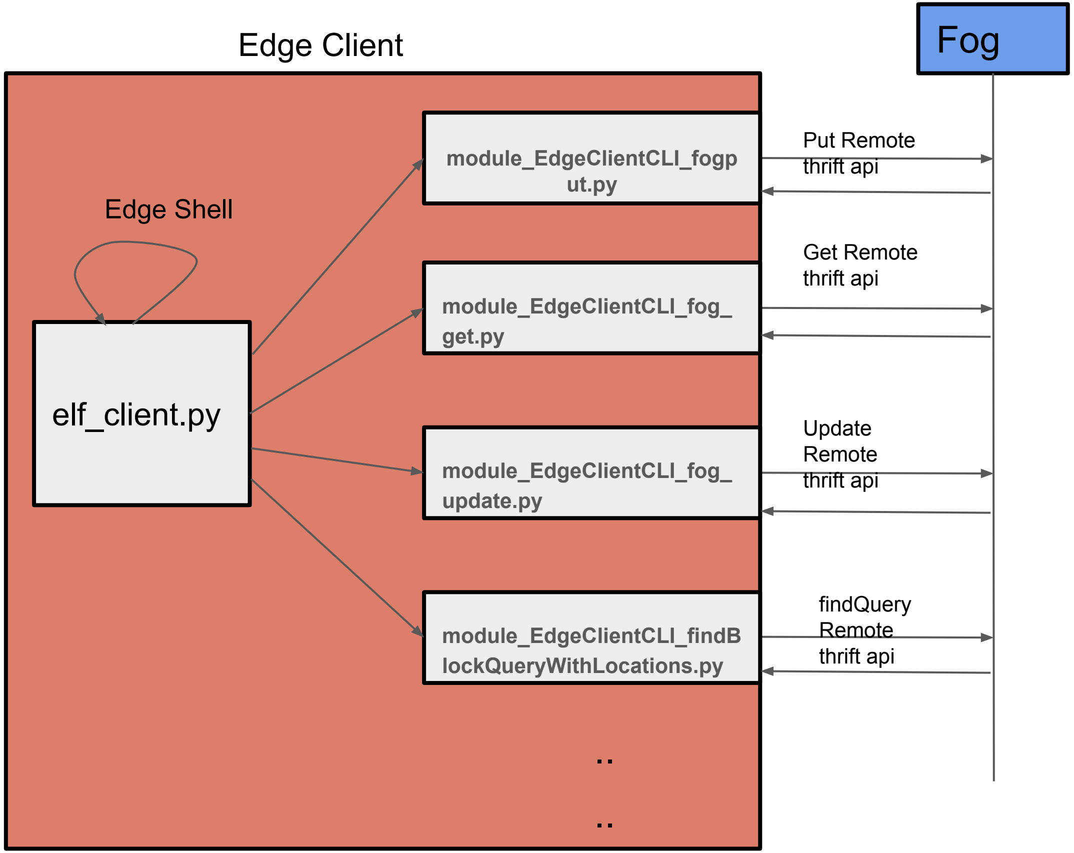
- The client code is written in Python
- It invokes the Server side apis.
- All the callable apis are present in **FogServices.thrift**
- For more information on how to add a Thrift Service Api check the link
  - https://thrift-tutorial.readthedocs.io/en/latest/usage-example.html
- Codebase has a command-line shell
- For examples of using it
  - Refer to **Readme.md**

Elfstore - client

- Major files are listed below
- **Elf_client.py** (main file, starting point)
- Api for each operation is added as a separate python module shown below:
- **module_EdgeClientCLI_fog_put.py**
- **module_EdgeClientCLI_fog_get.py**
- **module_EdgeClientCLI_fog_get.py**
- **module_EdgeClientCLI_findBlockQueryWithLocations.py**

Using the shell commands are listed in file

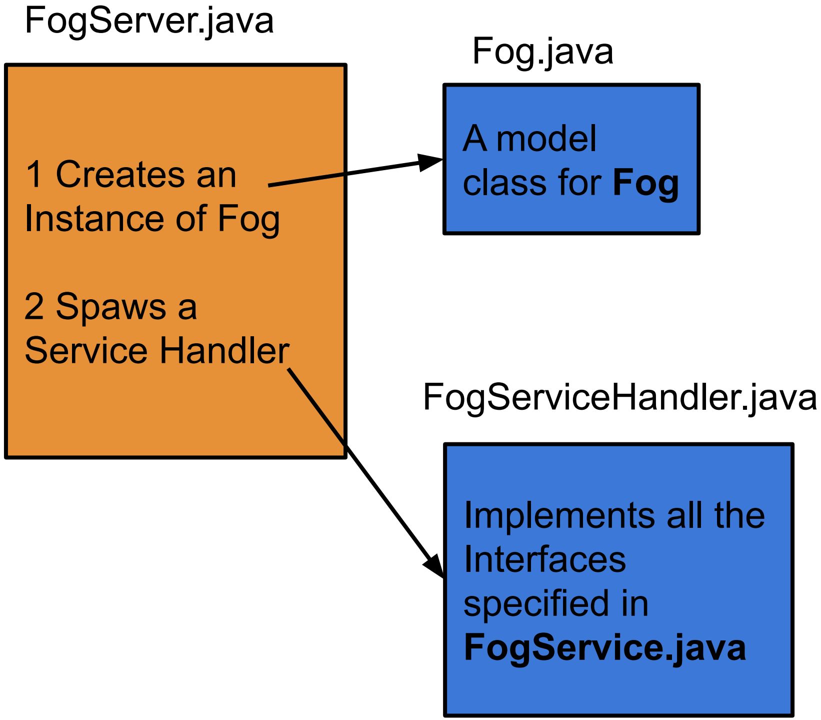
______


Edge Client
Fog
Edge Shell
elf_client.py
module_EdgeClientCLI_fogput.py
module_EdgeClientCLI_fog_get.py
module_EdgeClientCLI_fog_update.py
module_EdgeClientCLI_findBlockQueryWithLocations.py
..
..
Put Remote thrift api
Get Remote thrift api
Update Remote thrift api
findQuery Remote thrift api

Elfstore - Fog node Server

The main classes in Fog Node Server:

- **FogServer.java**
- **FogServiceHandler.java**
- **Fog.java**



FogServer.java

1 Creates an Instance of Fog

2 Spaws a Service Handler

Fog.java

A model class for **Fog**

FogServiceHandler.java

Implements all the Interfaces specified in **FogService.java**

FogServer.java

- **package com.dreamlab.edgefs.controlplane**
- This is the class which runs the Fog Server.
- Fog properties (such as IP, buddies, neighbors) are initialized from here.
- Creates a new **Fog** model instance.
- Creates the Fog, Neighbor and Buddy topology using the configuration file.
  - Cluster.conf (src/main/resources/cluster.conf)
- This class is also responsible for accessing the heartbeat exchange interval and many other properties
  - System.properties (src/main/resources/system.properties)
- Creates a FogServiceHandler instance.

Fog.java

- **package com.dreamlab.edgefs.controlplane**
- This is a model class which holds all the properties of a Fog
  - FogIP, FogID, FogPort, :BuddyID
  - List of all Neighbors, List of all Buddies
- Holds a lot of in-memory index structures such as
  - Block -> Location Map,
  - Metadata -> Block(s) Map
  - Edge -> Status Map
- Holds bloom-filter search index structures of Neighbors and Buddies
  - NeighborInfo (FogExchangeInfo Model class)
  - BuddyInfo (FogExchangeInfo Model class)
- Facilitates exchange of heartbeats between
  - Fog and Neighbor
  - Fog and Buddy
  - Fog and Edge

FogServiceHandler.java

- **package com.dreamlab.edgefs.servicehandler**
- Implements all the interfaces present in **FogService.java**
- It is the invocation point for all the Fog apis such as
  - put()
  - get()
  - update()
  - findBlockQuery()
- Makes use of the **Fog.java** instance to maintain data management.

GlobalReplicaAllocation.java

- **package com.dreamlab.edgefs.controlplane**
- Called during the ***put()*** api invocation
- Invoked by ***getWriteLocations()*** api
- Returns Fog nodes to which blocks will be written
- Is responsible for ***ensuring minimum reliability***.
- Ensures ***min replication*** and ***max replication***
- Chooses the Fog nodes using a ***differential replication scheme*** algorithm

LocalReplicaAllocation.java

- Performs the step of choosing an Edge to write the block
- Invoked by ***identifyLocalReplica()***
- Chooses the edge with a given ***WritePreference***
- ***identifyLocalReplica()*** Ensures that same edge is not chosen twice for a single block

EdgeServer.java

- **package com.dreamlab.edgefs.edge.server**
- This is the starting point for Edge Node.
- Initializes the **Edge** instance which is a model class for Edge Node
- Initializes the **EdgeServiceHandler.java** class which implements the interface **EdgeService.java**

Edge.java

- It is the model class for Edge Node
- **package com.dreamlab.edgefs.edge.model**
- Holds the **EdgeIP, EdgeID, EdgePort** etc.,
- Holds the info of **ParentFog IP, ParentFog port** etc.,
- Has apis to register to Fog
- Has apis to exchange heartbeats with the Parent Fog

EdgeServiceHandler.java

- package **com.dreamlab.edgefs.edge.handler**
- Implements the interface of **EdgeService.java**
- Consists of the apis such as
  - put()
  - get()
  - update()
- It is invoked as a remote thrift api by Fog to perform write/update/get of a block
- Also writes/updates the metadata index related to block

LocalStatsHandler.java

- **Package com.dreamlab.edgefs.misc**
- Considers max, min storage, max, min reliability of current Fog
- Computes the distribution of Edge devices into storage buckets of 4 types
  - HH, HL , LL, LH (Storage, Reliability)
  - Using equi-width and equi-depth partition

GlobalStatsHandler.java

- **Package com.dreamlab.edgefs.misc**
- Considers global max, min storage, max, min reliability of all Fogs
- Computes the distribution of Edge devices into storage buckets of 4 types
  - HH, HL , LL, LH (Storage, Reliability)
  - Using equi-width and equi-depth partition

package com.dreamlab.edgefs.model;

- Contains all the model class for data exchange between Fogs

package com.dreamlab.edgefs.iface;

- Contains all the following interfaces
  - Get Interfaces
  - Put Interfaces
  - Update Interfaces
  - Find Interfaces

package com.dreamlab.edgefs.writes

- Contains all the implementation for write and update operations.

package com.dreamlab.edgefs.reads

- Contains all the implementation for read operations.

package com.dreamlab.edgefs.thrift

- Contains all the thrift related java files, including model classes.
- Interfaces (FogService.java and EdgeService.java)

package com.dreamlab.edgefs.misc

- Contains BloomFilter related classes
- Data exchange format for exchanging bloomfilters between Fogs
- Contains CONSTANTS class which is a final static class.