PADMini Plugin Design Document

Sandipan Dey, Xianshu Zhu

September 17, 2010

1 Requirement Specification / User Experience

The following describes the basic overview of the requirements in terms of user experience:

- The user will first install the plugin from the PADMini website. Currently the plugin is implemented as chrome extension, but it should be developed as cross-browser plugin.
- When the user installs the plugin, the system will transparently (?) add the corresponding node to the underlying distributed network (to be implemented using open-chord).
- There can be two different implementations: the installer can be very light weight in the sense that it will install no component of the DHT in user machine and everything will be done through socket communication - this is typical client server implementation and since we are interested in P2P implementation, we do not want this. The alternative being that the DHT library will be installed to user's machine (as part of installation of plugin) so that any further insertion / removal of comments from his machine can be carried out without PADMini DDMServer's direct intervention (reducing load).
- Whenever the user adds / deletes a comment about an webpage he views, the corresponding entry will be updated in the DHT, using the webpage url as key and the user comment as value, i.e., from the user's clientside code we should be able to communicate to the underlying DHT (this may be done through applet code from javascript).
- Whenever a user visits a certain webpage, all the pre-existing comments from all users should be propagated to the list by a distributed query in the network.

2 Design

2.0.1 Communication to Underlying Network and Choice of DHT (Distributed Hash Table) algorithm

Chord, CAN, Pastry and Tapestry differ fundamentally only in the distributed routing and searching approach. They all rely on hash functions to map objects. However, they use different mapping mechanism. The lookup and space complexity is shown in table 1. We can use Chord as it is simplest to understand.

Protocols	lookup complexity	space complexity
Chord	O(log n)	O(log n)
Pastry	O(log n)	O(log n)
Tapestry	O(log n)	O(log n)

Table 1: Comparison

2.0.2 Node Failures

How Chord deals with nodes that fail or leave voluntarily. A basic ”stabilization” protocol is used to keep nodes' successor pointers up to date. The stabilization is to periodically verify node's immediate successor and tell the successor about the node. Moreover, it periodically refreshes the finger table entries.The corresponding function API is: *notify(int).*

2.0.3 Storage and Hash Function

Key can be the url of an webpage and value can be the the IP address list (separated by some sentinel value) from all users.

2.1 Plugin GUI

For chrome extension, html / css / javascript are used to design the GUI.

2.2 APIs

The following two methods are needed to be implemented for the plugin:

- GetExistingUserComments() : This method fetches all the existing comments about a given webpage and populates the drop down list in the plugin.
- SaveUserComments() : This methods saves the users comments in his local machine and simultaneously updates the DHT entries whenever necessary.

2.3 Security: Adding Captcha

While updating the finger table in the underlying chord network, we should safeguard ourself against attacks of the hackers / automated bots - captcha security is to be used for this purpose.

2.4 Architecture

The system has to be designed as multi-tier architecture. There are couple of design choices as explained below. But due to a bug in chrome the first approach with applets does not work from chroe extension, we shall go with the second one using websocket APIs.

2.4.1 Local Storage

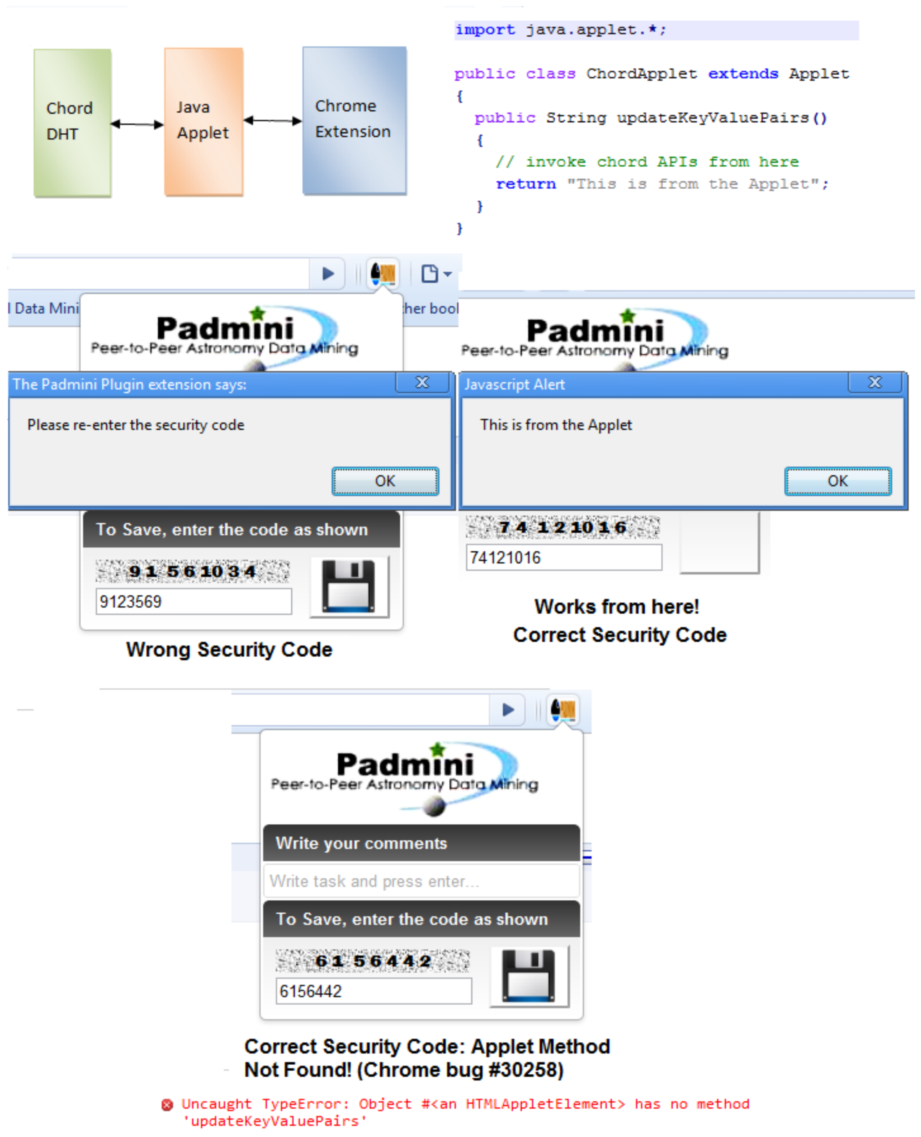
- Different Techniques exist for different browsers.
- Not all techniques are OK in terms of security.
- For IE ActiveX object for XML serialization works, but for no other browser.
- For Firefox DOM parser works, but allowing javascript to store file anywhere in filesystem is a potential security vulnerability issue.
- For Chrome, Ajax based XMLHttpRequest technique for XML serialization works only for remote files, Chrome does not allow javascript to write anywhere in the local filesystem. A workaround is Firefox native NPAPIs, but again with security vulnerabilities.
- Applet can be used as hack to write to local FS, but applet does not work for chrome extensions due to an existing bug.
- Chrome provides APIs for local storage / caching in a restricted way, need not write to any file, just use the browser local storage to persist user comments.

2.4.2 Get current Tab URL

- There exists chrome extension APIs to get the current window in the browser and selected tab in focus.
- For firefox, similar techniques needed to be investigated.

2.4.3 Key-Value pair Choice and Querying the DHT

- Key is the URL and the value can be either list of IPs of users or comments of users (compressed).


Chord DHT
Java Applet
Chrome Extension
Padmini
Peer-to-Peer Astronomy Data Mining
OK
Wrong Security Code
Works from here!
Correct Security Code
Correct Security Code: Applet Method Not Found! (Chrome bug #30258)

```
import java.applet.*;

public class ChordApplet extends Applet
{
  public String updateKeyValuePairs()
  {
    // invoke chord APIs from here
    return "This is from the Applet";
  }
}
```

The Padmini Plugin extension says:

Please re-enter the security code

To Save, enter the code as shown

9 1 5 6 10 3 4

9123569

Javascript Alert

This is from the Applet

7 4 1 2 10 1 6

74121016

Write your comments

Write task and press enter...

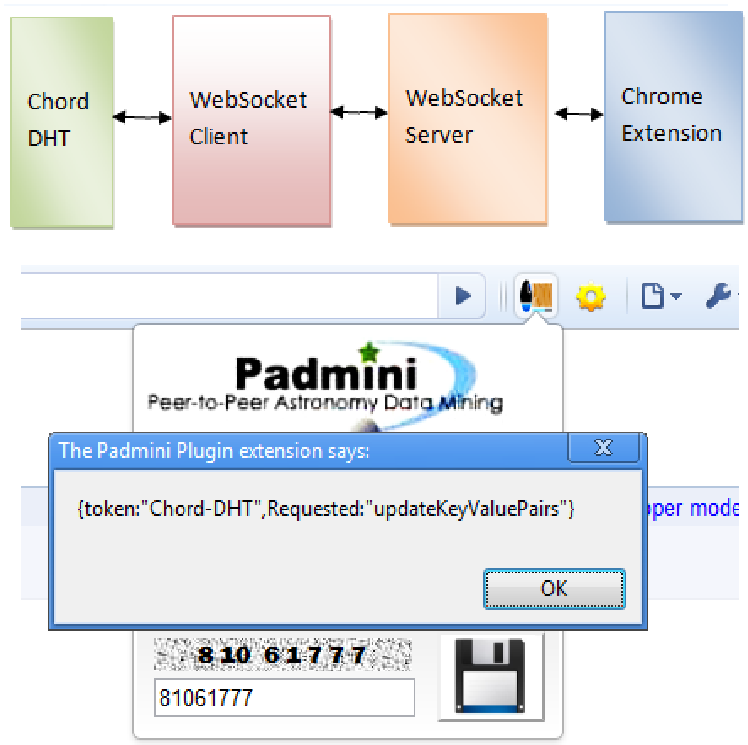
To Save, enter the code as shown

6 1 5 6 4 4 2

6156442

```
Uncaught TypeError: Object #<an HTMLAppletElement> has no method
'updateKeyValuePairs'
```

Figure 1: Design 1 (with Backend Applet): Chrome Extension with Captcha Validation



The Padmini Plugin extension says:

```
{token:"Chord-DHT",Requested:"updateKeyValuePairs"}
```

81061777

Figure 2: Design 2 (with Websocket APIs): Chrome Extension with Captcha Validation

- If the values are IPs of users, after DHT query for finding user comments will return a list of IPs (NOT comments), in that case the installer should simultaneously install a serversocket in each node that will respond to the subsequent query for finding the comments from the corresponding IPs.
- If the values are comments of users, DHT query is easy.
- Complexity of implementation vs. scalability tradeoff.
- While saving an item DHT query for saving needs to be fired only when there is no local storage for the corresponding URL.

3 Issues that need to be discussed

3.1 Node join or leave / Overlay vs physical network maintenance

The main problem of DHT structure is that the maintenance of DHT mechanism is complex. The frequent joining in and exiting will increase the cost of maintainace. The structured P2P system des not adapt to the highly adaptive internet environment.

- For our application senarior, user leave the network when he close the browser and join the network when he open the browser. Will this join and leave activity be too frequent? For BitTorrent, user leave or join the network depends on the user shut down or open the software. In the same way, can we ask user keep opening a particular web page to maintain its appearance in the p2p network?
- An alternate option is to start a service that can run in the background. This service is not depend on the browser.

3.2 Plugin Interface

- How we design the interface, when a web page contains multiple places that allow users to add comments.

3.3 Connect Plugin with OpenChord

Connect javascript with java using applet / use websocket APIs?

4 Snapshots

The following represents the snapshots of the system at different stages.

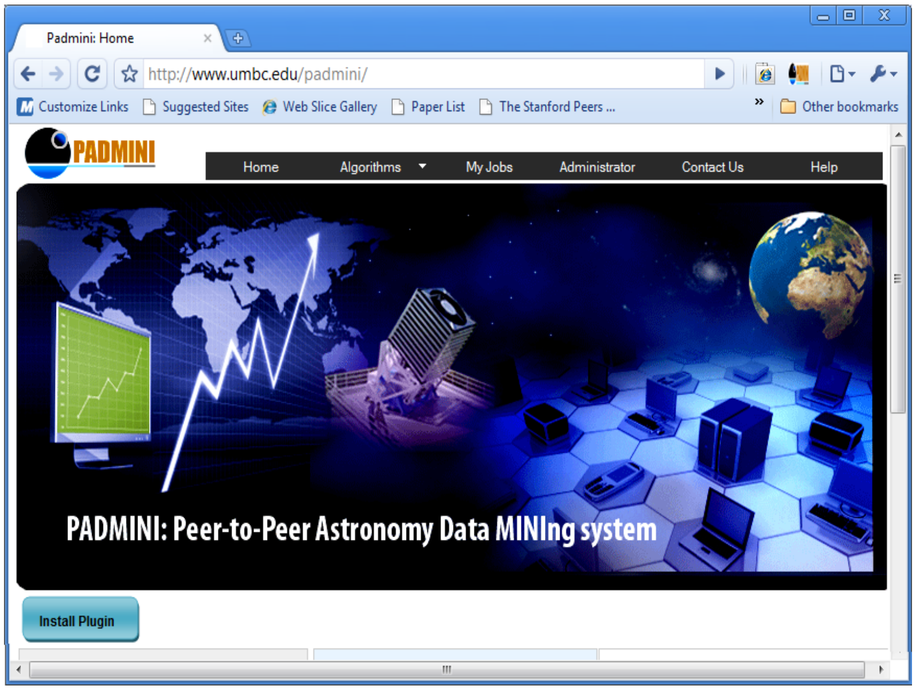


Figure 3: Installing the plugin as Chrome Extension

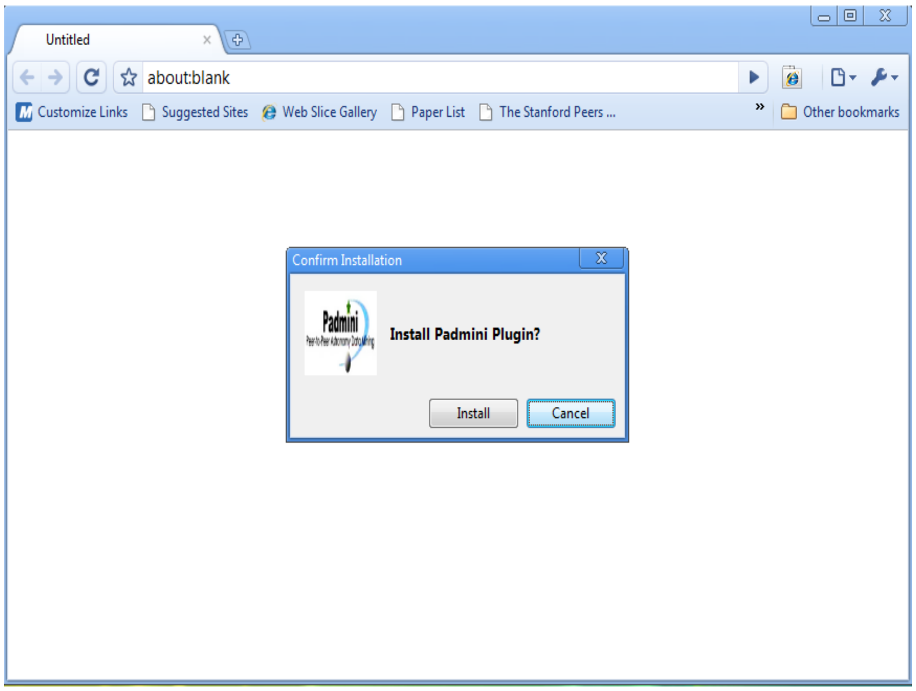
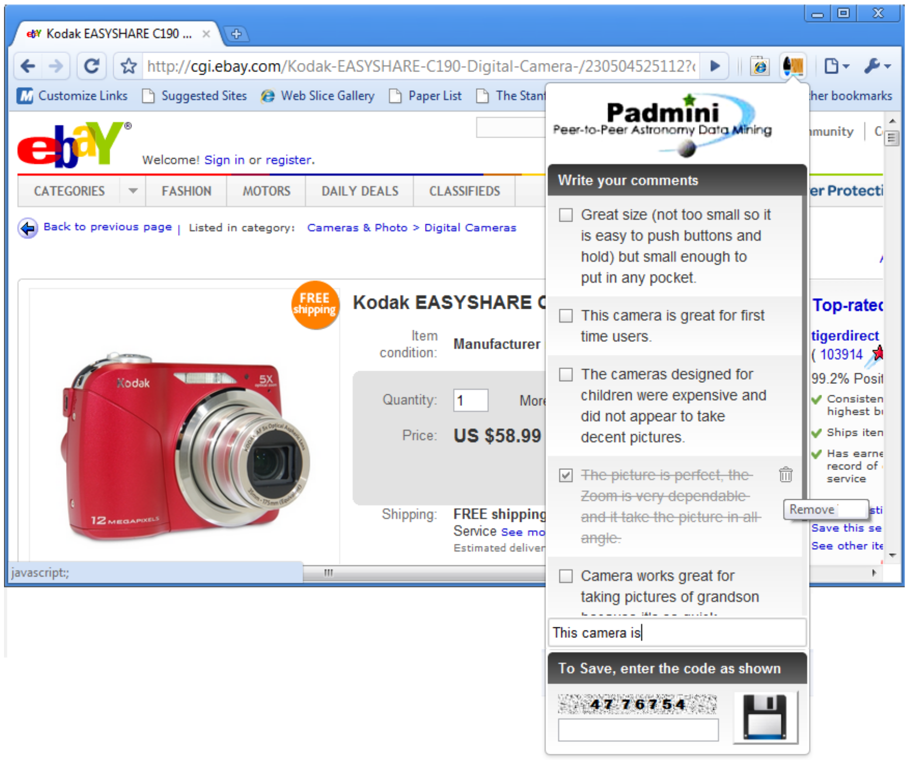


Figure 4: Installing the plugin as Chrome Extension



Write your comments

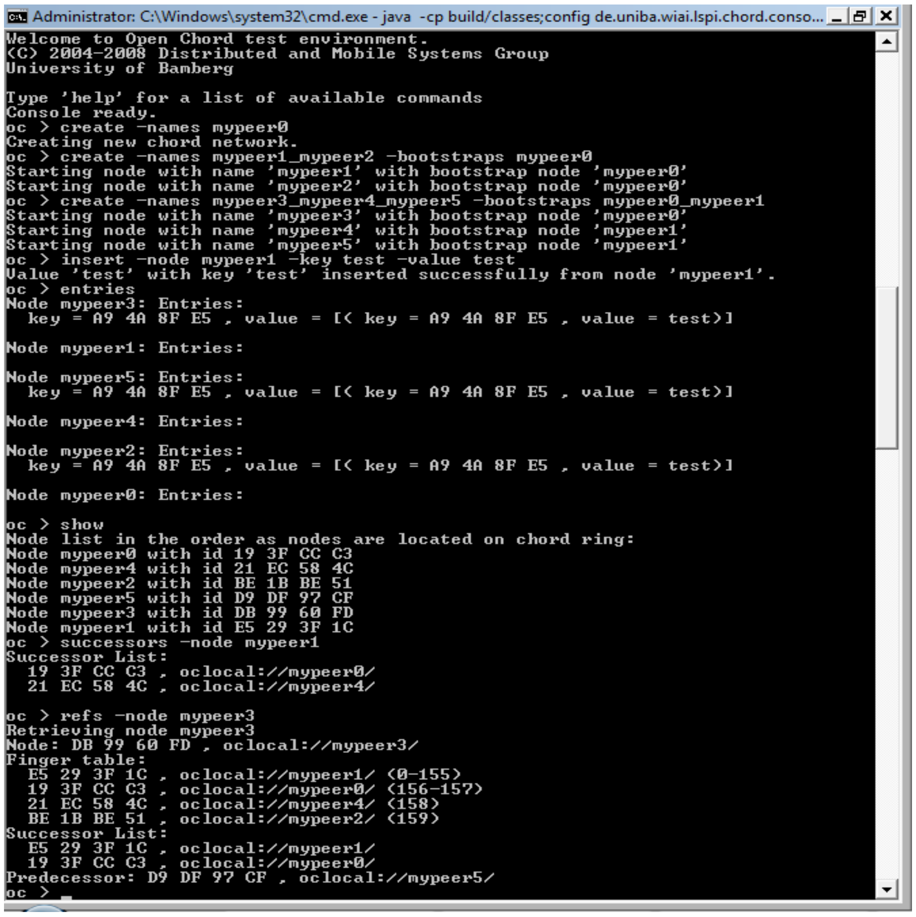
- [ ] Great size (not too small so it is easy to push buttons and hold) but small enough to put in any pocket.
- [ ] This camera is great for first time users.
- [ ] The cameras designed for children were expensive and did not appear to take decent pictures.
- [x] ~~The picture is perfect, the Zoom is very dependable and it take the picture in all angle.~~
- [ ] Camera works great for taking pictures of grandson

This camera is

To Save, enter the code as shown

4776754

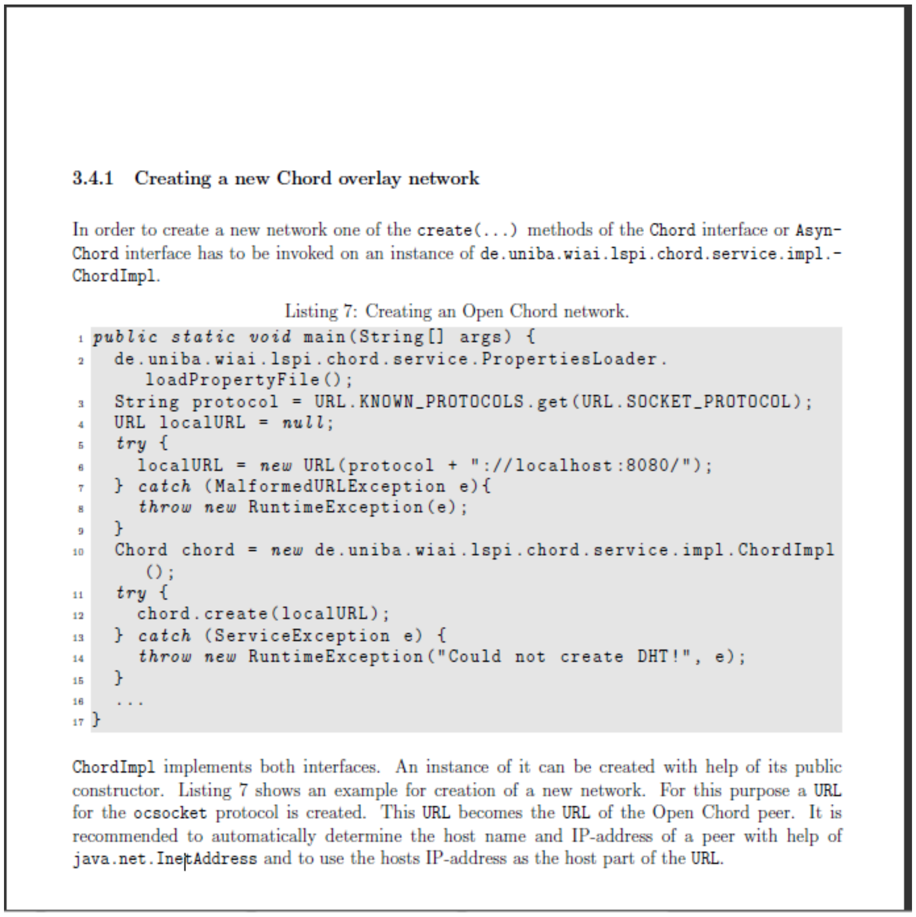
Figure 5: Writing Comments with the Chrome Extension with Captcha Validation



Administrator: C:\Windows\system32\cmd.exe - java -cp build/classes;config de.uniba.wiai.lspi.chord.conso...

```
Welcome to Open Chord test environment.
(C) 2004-2008 Distributed and Mobile Systems Group
University of Bamberg

Type 'help' for a list of available commands
Console ready.
oc > create -names mypeer0
Creating new chord network.
oc > create -names mypeer1_mypeer2 -bootstraps mypeer0
Starting node with name 'mypeer1' with bootstrap node 'mypeer0'
Starting node with name 'mypeer2' with bootstrap node 'mypeer0'
oc > create -names mypeer3_mypeer4_mypeer5 -bootstraps mypeer0_mypeer1
Starting node with name 'mypeer3' with bootstrap node 'mypeer0'
Starting node with name 'mypeer4' with bootstrap node 'mypeer1'
Starting node with name 'mypeer5' with bootstrap node 'mypeer1'
oc > insert -node mypeer1 -key test -value test
Value 'test' with key 'test' inserted successfully from node 'mypeer1'.
oc > entries
Node mypeer3: Entries:
  key = A9 4A 8F E5 , value = [( key = A9 4A 8F E5 , value = test)]

Node mypeer1: Entries:

Node mypeer5: Entries:
  key = A9 4A 8F E5 , value = [( key = A9 4A 8F E5 , value = test)]

Node mypeer4: Entries:

Node mypeer2: Entries:
  key = A9 4A 8F E5 , value = [( key = A9 4A 8F E5 , value = test)]

Node mypeer0: Entries:

oc > show
Node list in the order as nodes are located on chord ring:
Node mypeer0 with id 19 3F CC C3
Node mypeer4 with id 21 EC 58 4C
Node mypeer2 with id BE 1B BE 51
Node mypeer5 with id D9 DF 97 CF
Node mypeer3 with id DB 99 60 FD
Node mypeer1 with id E5 29 3F 1C
oc > successors -node mypeer1
Successor List:
  19 3F CC C3 , oclocal://mypeer0/
  21 EC 58 4C , oclocal://mypeer4/

oc > refs -node mypeer3
Retrieving node mypeer3
Node: DB 99 60 FD , oclocal://mypeer3/
Finger table:
  E5 29 3F 1C , oclocal://mypeer1/ (0-155)
  19 3F CC C3 , oclocal://mypeer0/ (156-157)
  21 EC 58 4C , oclocal://mypeer4/ (158)
  BE 1B BE 51 , oclocal://mypeer2/ (159)
Successor List:
  E5 29 3F 1C , oclocal://mypeer1/
  19 3F CC C3 , oclocal://mypeer0/
Predecessor: D9 DF 97 CF , oclocal://mypeer5/
oc > _
```

Figure 6: The Open Chord Console: Simulating a Chord overlay network in one JVM using Open-Chord



3.4.1 Creating a new Chord overlay network

In order to create a new network one of the `create(...)` methods of the `Chord` interface or `AsynChord` interface has to be invoked on an instance of `de.uniba.wiai.lspi.chord.service.impl.ChordImpl`.

Listing 7: Creating an Open Chord network.

```
public static void main(String[] args) {
  de.uniba.wiai.lspi.chord.service.PropertiesLoader.
     loadPropertyFile();
  String protocol = URL.KNOWN_PROTOCOLS.get(URL.SOCKET_PROTOCOL);
  URL localURL = null;
  try {
    localURL = new URL(protocol + "://localhost:8080/");
  } catch (MalformedURLException e){
    throw new RuntimeException(e);
  }
  Chord chord = new de.uniba.wiai.lspi.chord.service.impl.ChordImpl
     ();
  try {
    chord.create(localURL);
  } catch (ServiceException e) {
    throw new RuntimeException("Could not create DHT!", e);
  }
  ...
}
```

`ChordImpl` implements both interfaces. An instance of it can be created with help of its public constructor. Listing 7 shows an example for creation of a new network. For this purpose a `URL` for the `ocsocket` protocol is created. This `URL` becomes the `URL` of the Open Chord peer. It is recommended to automatically determine the host name and IP-address of a peer with help of `java.net.InetAddress` and to use the hosts IP-address as the host part of the `URL`.

Figure 7: The Open Chord API: Creating Network

5 Tasks

5.1 Frontend Tasks

5.1.1 GUI Design of for Chrome Extension

Task	Status	ETA	Assigned To
Basic html/css/js/json implementation	done		Sandipan
Improving page design (e.g. adding index pages instead of just scroll bar)	yet to be done	end of September	Sandipan

5.2 Middleend Tasks

Task	Status	ETA	Assigned To
Basic websocket implementation to call to backend java APIs for chord	done		Sandipan
Implement save locally in web cache	done		Sandipan
Implement save in DHT	in progress	end of September	Sandipan
Implement retrieve from DHT	in progress	end of September	Sandipan
Implement delete locally	done		Sandipan
Implement delete from DHT	in progress	end of September	Sandipan
Implement calls to chord DHT APIs through websocket	in progress	end of September	Sandipan

5.3 Backend Tasks

Task	Status	ETA	Assigned To
Basic investigation with the open-chord APIs	done		Sandipan
Implement save / retrieve / delete in Chord	done		Xianshu