Telemetry Summery

**1. The database structure of Dual SIM Card**

The structure of network and icc are changed from object to array, the element of array is the same as the former object.

The former:

"network": {

"mcc": "216",

"mnc": "70",

"operator": "vodafone HU vodafone"

},

"icc": {

"spn": "vodafone",

"mcc": "216",

"mnc": "70"

},

After the change:

"network": [

{

"mcc": "216",

"mnc": "70",

"operator": "vodafone HU vodafone"

},

{

"mcc": "216",

"mnc": "70",

"operator": "vodafone HU vodafone"

}

],

"icc": [

{

"spn": "vodafone",

"mcc": "216",

"mnc": "70"

},

{

"spn": "vodafone",

"mcc": "216",

"mnc": "70"

}

]

**2. The rule of sending message: 1) Retry 2) Monthly**

The entry point of FtuPing is ensurePing, after promise chain of self.initSettings(), self.initPreinstalledApps(), self.initImeis() are finished, self.startPing() is called. An interval of this.\_pingTimer is set through the code:

this.\_pingTimer = setInterval(this.tryPing.bind(this), this.\_tryInterval);

As a result this.tryPing() is call by interval of this.\_tryInterval.

Notice: Even this.\_networkFailCount >= this.\_maxNetworkFails and Max voice network failures reached, pinging anyway!

This.ping() is called in function tryPing() to send data collected above to server by TelemetryRequest.

this.\_pingEnabled is used to indicate ping or not, it is set to true by default. After ajax callback function of pingSuccess is called, this.\_pingEnabled is set to false, and ping action will stoped at funciton startPing(). The function of reset can force set this.\_pingEnabled to true.

As a summery, retry of action ping is set by setInterval in function starting, as the value of this.\_tryInterval is 60\*60\*1000, ping is activated per hour. This.\_networkFailCount is used to count the fail times of checkMobileNetwork , even it is bigger than this.\_maxNetworkFails, ping action will continue until success, once success, the ping action will stop. You can set this.\_pingEnabled to true by function reset call, then ping action will consume. The mechanism of Monthly call does note exits in current code.

**3. Which service enable Telemetry(The start of Telemetry)**

3.1 introduce to System App

FTU is known as First Time Usage, a part of System Application. System is a base Application, and all the related files are located in gaia/apps/system/. Some startup js files are launched by index.html, including js/applications.js, shared/lazy\_loader/lazy\_loader.js, js/service.js, js/base\_module.js, js/app.js, js/bootstrap.js, and they are very important for System App startup. Other js files needed are loaded via lazy\_loader.js if needed.

3.2 Concept of BaseModule

Window.BaseModule which is export in file js/base\_module.js is a very import object for module management in System App.

BaseModule is a class skeleton which helps you to build a module with

\* Centralized event handler

\* Centralized settings observation

\* Sub modules management including loading and starting

\* Import preload files

\* DOM rendering

\* Consistent logging function with System boot time and module name

\* Common publishing interface

BaseModule is a very important concept used to warp all services and modules used in System Application, Launcher, FtuLauncher, and Core are all managed in the schdule of BaseModule.

Important concepts of BaseModule are create, instantiate, and SUB\_MODULES.

BaseModule.create is used to create a module based on base module and give properties. A constructor will be returned and placed in AVAILABLE\_MODULES if you specify an unique name in the prototype.

BaseModule.instantiate is used to instantiate an object by name through the constructor stored in AVAILABLE\_MODULES.

BaseModule.SUB\_MODULES is an array store the sub modules belong to this module. BaseModule will load and then start these sub modules automatically.

3.3 How FtuPing is started?

js/bootstrap.js

js/app.js

Core(js/core.js)

AppCore(js/app\_core.js)

AppWindowManager(js/app\_window\_manager.js)

FtuLauncher(js/ftu\_launcher.js)

**4. what is the process when the data can not be gotten**

Some asynchronous data are got through window.asyncStorage, the following example showing how data is got through window.ayncStorage by itemKeys.

function getAsyncStorageItems(itemKeys, callback) {

var itemsLeft = itemKeys.length;

var items = {};

itemKeys.forEach(function(key) {

window.asyncStorage.getItem(key, function(value) {

itemsLeft--;

items[key] = value;

if (itemsLeft === 0 && callback) {

callback(items);

}

});

});

}

Some data about Setting is get through navigator.mozSettings, in the form of below.

function self\_getSettings(settingKeys, callback) {

var settingsLeft = settingKeys.length;

var settings = {};

var lock = window.navigator.mozSettings.createLock();

settingKeys.forEach(function(key) {

var request = lock.get(key);

request.onsuccess = function(evt) {

var value = request.result[key];

settingsLeft--;

settings[key] = value;

if (settingsLeft === 0 && callback) {

callback(settings);

}

};

});

}

So the case of failing to get a data is handled by window.ayncStorage or navigator.mozSettings, and has not direct relation with FtuPing.

**5. Change FMD and Kill-Switch to Anti-Theft**

**6. Confirm the content of URL and json**

{

"activationTime": 1398300000000,

"devicePixelRatio": 1,

"deviceinfo.MEID": [123456789012345, 123456789012345],

"deviceinfo.firmware\_revision": "",

"deviceinfo.hardware": "qcom",

"deviceinfo.last\_updated": "",

"deviceinfo.os": "2.0.0.0-prerelease",

"deviceinfo.platform\_build\_id": "20150817143800",

"deviceinfo.platform\_version": "37.0",

"deviceinfo.product\_model": "go\_flip",

"deviceinfo.software": "Boot2Gecko 2.2.0.0-prerelease",

"deviceinfo.update\_channel": "default",

"findmydevice.enabled": "",

"icc": [

{

"spn": "vodafone",

"mcc": "216",

"mnc": "70"

},

{

"spn": "vodafone",

"mcc": "216",

"mnc": "70"

}

],

"network": [

{

"mcc": "216",

"mnc": "70",

"operator": "vodafone HU vodafone"

},

{

"mcc": "216",

"mnc": "70",

"operator": "vodafone HU vodafone"

}

],

"locale": "hu",

"pingID": "8f1bbb30-a1f0-4403-938f-88acf19df5ad",

"preinstalled": {

app://deviceinfo.h5os/manifest.webapp:"DeviceInfo",

http://mochi.test:8888/manifest.webapp:"Mochitest"

}

"info": {

"appUpdateChannel": "default",

"reason": "ftu",

"appVersion": "31.0a1",

"geoCountry": "HU",

"appName": "FirefoxOS",

"appBuildID": "20140423185429"

},

"pingTime": 1398300000000,

"screenWidth": 320,

"screenHeight": 480,

"ver": 2

}