

# Creating Synergies on the Cloudbox

Mohan Sundar  
Samsung Electronics

**TIZEN™**  
**DEVELOPER  
SUMMIT**  
2015 BENGALURU   
JULY 30-31, THE RITZ-CARLTON

# Index



# Index

- **Background**
- **What's Cloudbox?**
- **How to use Cloudbox?**
- **Cloudbox Internals**
- **Sample App (Unified Single View)**
- **Summary**
- **Q&A**

Background



# Cloud Storages – Advantages

- **Cloud (Storage) = “Remote” + Storage**
- User can use Cloud Storages **from any devices**
- User can **easily access** Cloud Services from mobile
- Supports **free storage** limit for special period
- User can **save space** in Device



# Cloud Storages – Limitations

- We must **install the storage application** to access the contents
- Difficult to manage the contents stored in **multi clouds**
- Can't view all the multi cloud content using **single view**

And Then ...

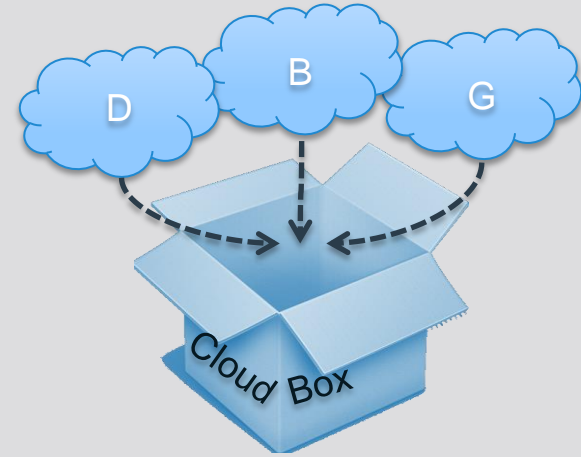
- We Can give you the Solutions through this Cloudbox Session

What's Cloudbox?



# Cloudbox ?!

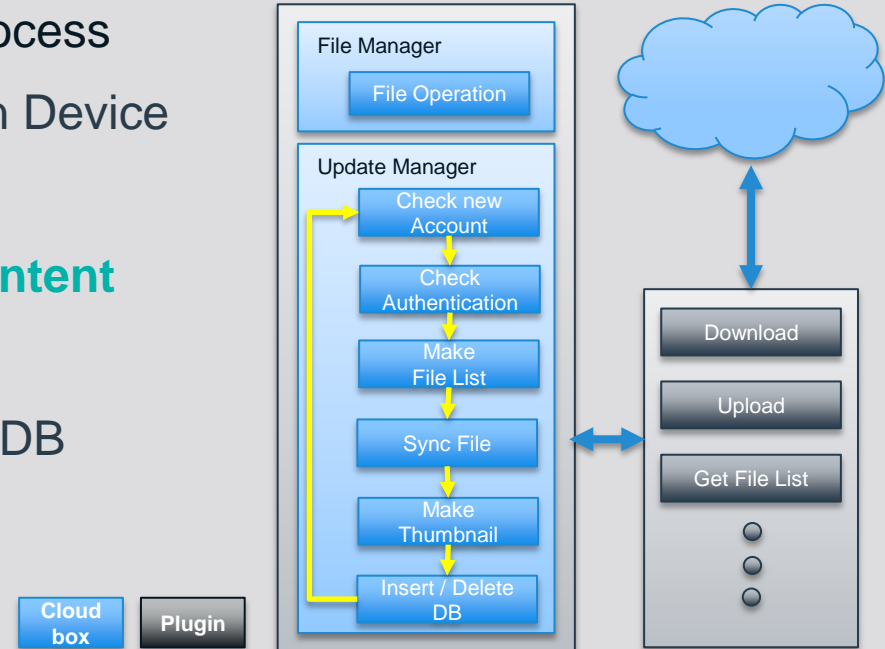
- **Cloudbox** = Clouds in Box
- Inbuilt within **Tizen Platform**
- **Uniform Interface** to access multi clouds
- **Unified single view** for all multi cloud contents
- **Plug-In/Plug-Out** your preferred service on demand





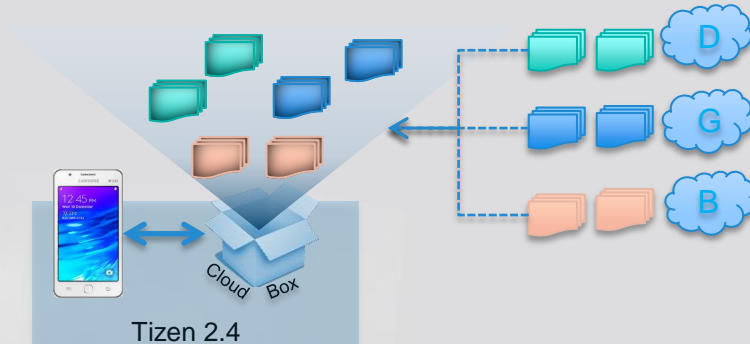
# The Role of Cloudbox

- In Platform Side, Cloudbox manage the Contents of Clouds
  - Manages **User Authentication** process
  - Creates **File Tree & Thumbnail** on Device
  - Supports **File Operation** for Cloud
  - Periodically, **Synchronizes the content** in device and cloud
  - **Insert / Delete** the Content Info to DB



# Cloudbox benefits..

- **Access cloud content like local content**
  - Require very minimal changes in your code
- **User gets access to all his cloud content in one place**
  - You can take advantage of this convenience, Provide new service using it
  - Cloudbox can improve your application popularity & download ratio



Scope to develop exciting new features..  
More happy users!!!



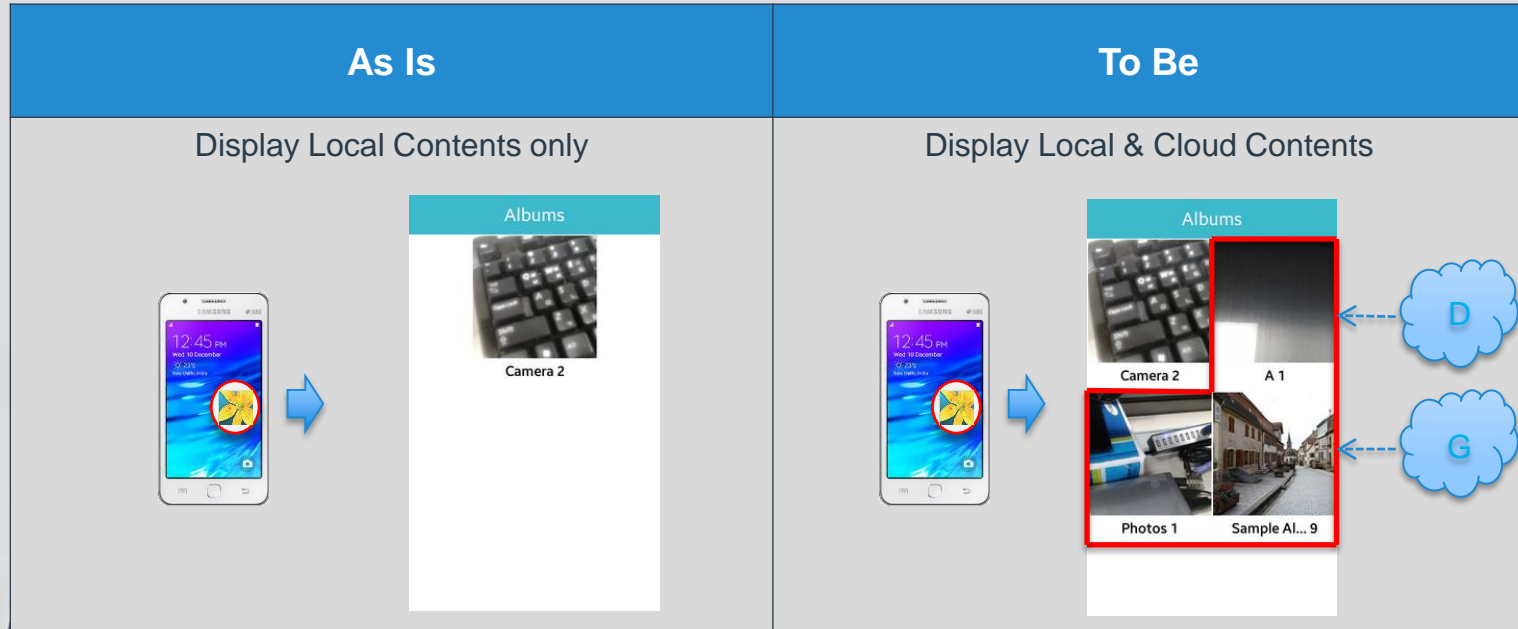
Rule based  
content backup

Free quota  
management  
across cloud

....

# Cloudbox benefits.. (contd.)

- Open the Gallery(Reference Application) in Tizen 2.4



Apply Cloudbox



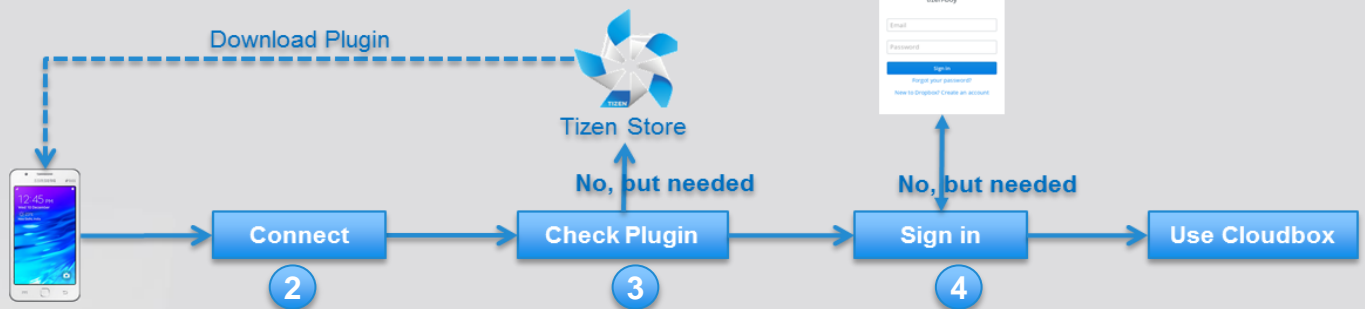
# How to use Cloudbox in your Application?

4 simple steps

- 1 Visit [developer.tizen.org](http://developer.tizen.org)  
Get familiar with Cloudbox APIs
- 2 Register your application with Cloudbox
- 3 Download desired storage plugins, if not available
- 4 Notify the user to 'Sign-in' if he don't have access yet



1



# Search the APIs for Cloudbox

Step 1

- Since tizen 2.4, Cloudbox is released  
So You can find the APIs in [Tizen Developer Site](#)

▼ Native API Reference

- Application Framework
- ▶ Base
- Content
- Context
- Location
- Messaging
- ▶ Multimedia
- ▶ Network
- ▶ Security
- ▼ Social
  - ▼ Service Adaptor
  - ▼ Service Plugin
  - Storage
- System
- Telephony

## Service Adaptor

Social

### Functions

int	<b>service_adaptor_create</b> (service_adaptor_h *service_adaptor)	Create Service Adaptor.
int	<b>service_adaptor_destroy</b> (service_adaptor_h service_adaptor)	Destroy Service Adaptor.
int	<b>service_adaptor_foreach_plugin</b> (service_adaptor_h service_adaptor, service_adaptor_plugin_cb callback, void *user_data)	Foreach the list of plugin.
int	<b>service_adaptor_query_plugin_by_file</b> (service_adaptor_h service_adaptor, const char *file_path, char **plugin_uri)	Retrieve an plugin URI with local file path.
int	<b>service_adaptor_create_plugin</b> (service_adaptor_h service_adaptor, const char *plugin_uri, service_plugin_h *plugin)	Create service plugin handle.
int	<b>service_adaptor_get_last_result</b> (int *err)	Gets service specific last result.
int	<b>service_adaptor_get_last_error_message</b> (char **message)	Gets service specific last result error message.

### Typedefs

typedef struct _service_adaptor_s *	<b>service_adaptor_h</b>	The handle for connection and managing plugin handle of Service Adaptor.
typedef bool(*	<b>service_adaptor_plugin_cb</b> )(char *plugin_uri, int service_mask, void *user_data)	Callback for service_adaptor_foreach_plugin API.

# Register with Cloudbox

Step 2

- By using below API(**service\_adaptor\_create**)

```
/**
 * @brief Create Service Adaptor
 * @since_tizen 2.4
 *
 * @param[out] service_adaptor The Service Adaptor handle
 * @remarks    @a service_adaptor must be released memory using service_adaptor_destroy(), when a program no longer
 needs any function of Service Adaptor
 * @see        service_adaptor_destroy()
 * @return 0 on success, otherwise a negative error value
 * @retval #SERVICE_ADAPTOR_ERROR_NONE Successful
 * @retval #SERVICE_ADAPTOR_ERROR_INVALID_PARAMETER Invalid parameter
 * @retval #SERVICE_ADAPTOR_ERROR_UNKNOWN Unknown error
 */
int service_adaptor_create(service_adaptor_h *service_adaptor);
```

# Check Plugins

Step 3

- By using below API(`service_adaptor_foreach_plugin`)

```
/**
 * @brief Foreach the list of plugin
 * @details Iterate to all installed plugin
 * @since_tizen 2.4
 *
 * @param[in] service_adaptor The handle of Service Adaptor
 * @param[in] callback The callback for foreach plugin
 * @param[in] user_data Passed data to callback
 * @return 0 on success, otherwise a negative error value
 * @retval #SERVICE_ADAPTOR_ERROR_NONE Successful
 * @retval #SERVICE_ADAPTOR_ERROR_NO_DATA There is no available plugins
 * @retval #SERVICE_ADAPTOR_ERROR_INVALID_PARAMETER Invalid parameter
 * @retval #SERVICE_ADAPTOR_ERROR_UNKNOWN Unknown error
 * @pre @a service_adaptor must be issued by service_adaptor_create()
 */
int service_adaptor_foreach_plugin(service_adaptor_h service_adaptor,
                                   service_adaptor_plugin_cb callback,
                                   void *user_data);
```



# Check “Sign in” status

Step 4

- By using below API(`service_plugin_start`)

```
/**
 * @brief Requests start initialization for service plugin
 * @since_tizen 2.4
 *
 * @param[in] plugin          The handle for use Plugin APIs
 * @param[in] service_mask    The flag for use service plugins, this flag can be masked multiple enum
 *                             (#service_plugin_service_type_e)
 * @return 0 on success, otherwise a negative error value
 * @return If return value is #SERVICE_ADAPTOR_ERROR_NOT_AUTHORIZED, request signup to authorization application
 * @retval #SERVICE_ADAPTOR_ERROR_NONE Successful
 * @retval #SERVICE_ADAPTOR_ERROR_INVALID_PARAMETER Invalid parameter
 * @retval #SERVICE_ADAPTOR_ERROR_NOT_AUTHORIZED Need authorization
 * @retval #SERVICE_ADAPTOR_ERROR_TIMED_OUT Timed out
 * @retval #SERVICE_ADAPTOR_ERROR_IPC_UNSTABLE IPC failed with Service Adaptor Daemon
 * @retval #SERVICE_ADAPTOR_ERROR_UNKNOWN Unknown error
 */
int service_plugin_start(service_plugin_h plugin,
                        int service_mask);
```

# But, How to Download & Upload ?

Ready!!

- You just use the **Same POSIX API** for Local & Cloud
  - open / read / write / close ...
- **And What?**
  - Cloudbox will perform all file operations about Cloud instead of you
  - Wow, Simple & Easy!!

# Cloudbox Operation



# How Support the POSIX API?

- In Cloudbox, Support below **File Operations** by using FUSE

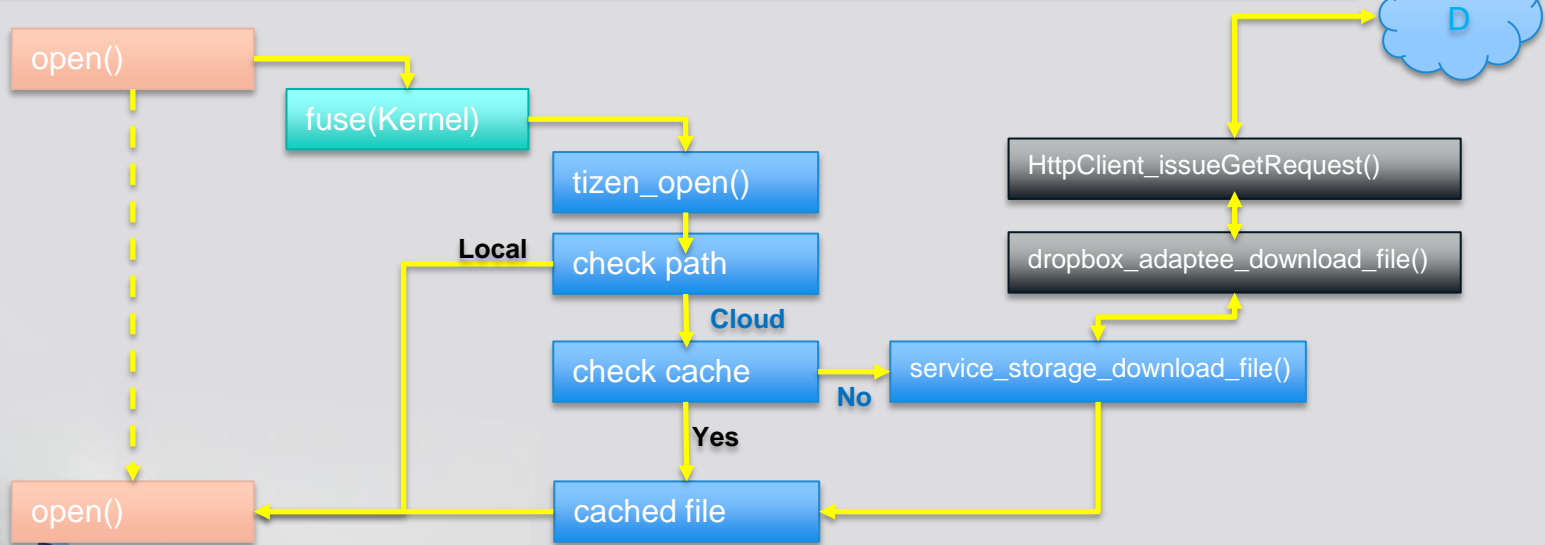
- + tizen\_getattr()
- + tizen\_access()
- + tizen\_readlink()
- + tizen\_readdir()
- + tizen\_mknod()
- + tizen\_mkdir()
- + tizen\_symlink()
- + tizen\_unlink()
- + tizen\_rmdir()
- + tizen\_rename()
- + tizen\_link()
- + tizen\_chmod()
- + tizen\_chown()
- + tizen\_truncate()

- + tizen\_open()
- + tizen\_read()
- + tizen\_write()
- + tizen\_statfs()
- + tizen\_release()
- + tizen\_fsync()
- + tizen\_flush()
- + tizen\_fallocate()
- + tizen\_setxattr()
- + tizen\_getxattr()
- + tizen\_listxattr()
- + tizen\_removexattr()
- + tizen\_utimens()

# Operation for “open”

- If You use “open” for Cloud Contents...

```
fd = open("/opt/storage/tnfs/cloud/Dropbox/Z1.jpg, 'rb');
```



# Operation for “rename”

App

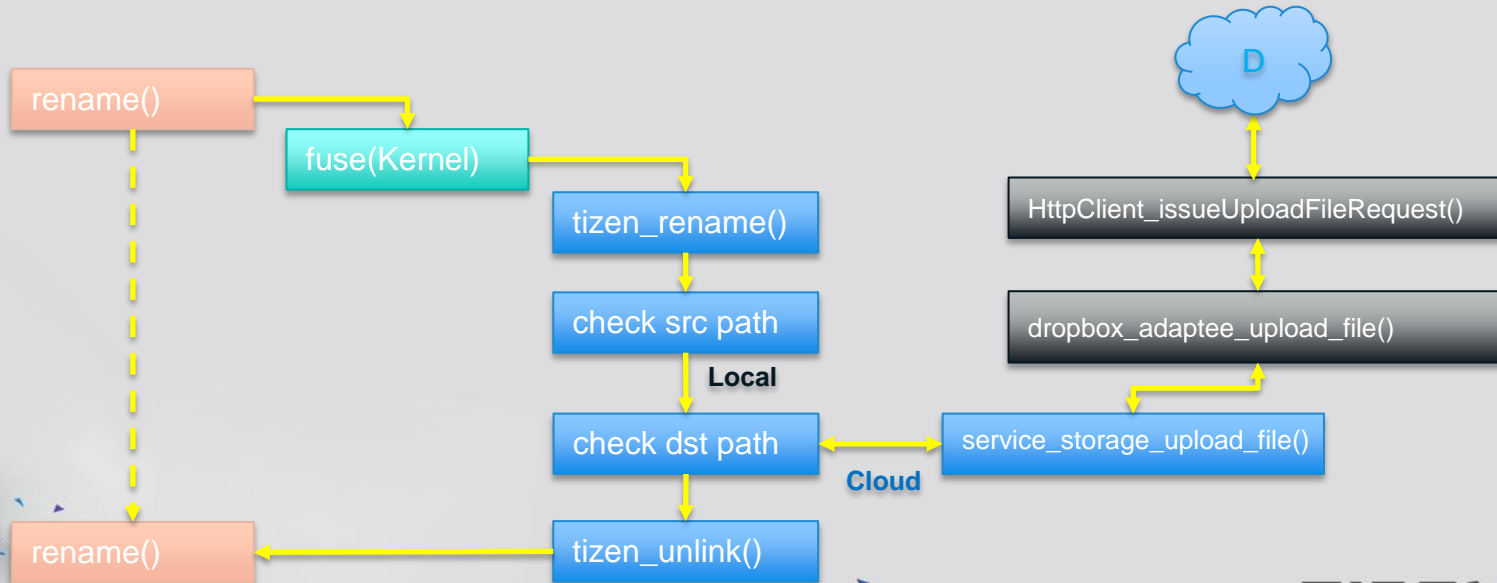
Kernel

Cloud  
box

Plugin

- Move a local content to Cloud using “rename”

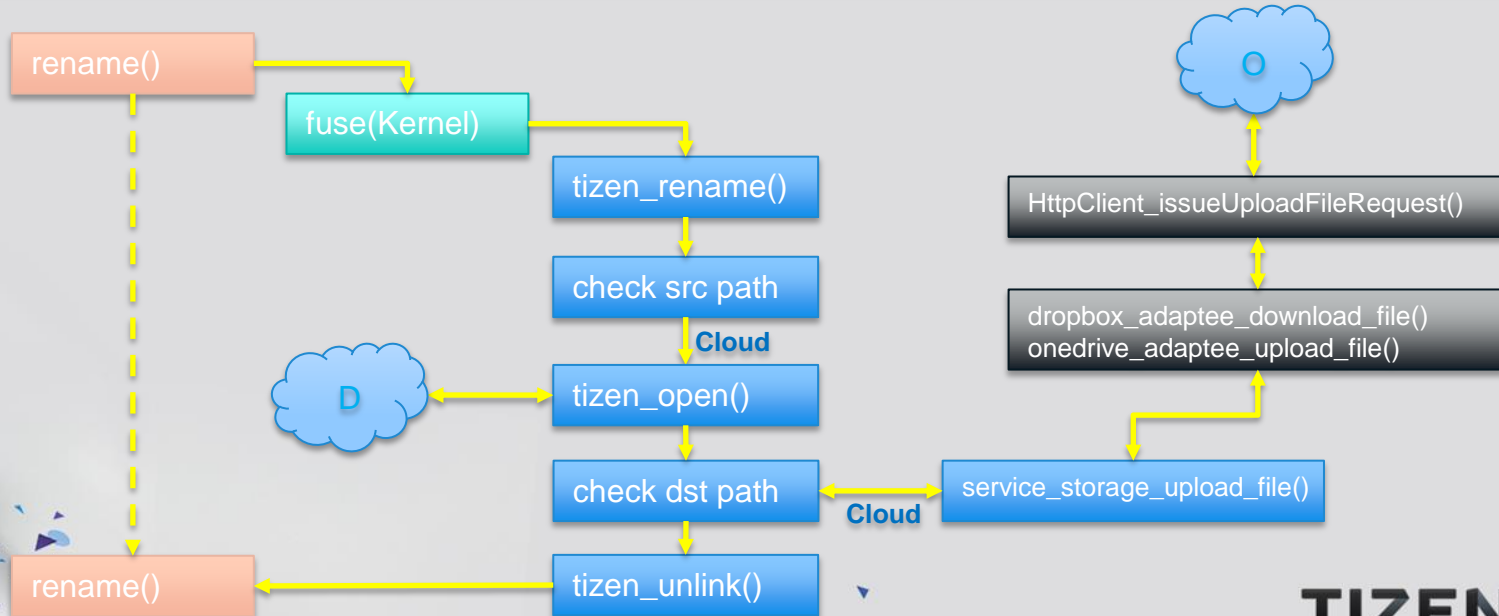
```
rename("/opt/usr/media/Image/TDS.jpg", "/opt/storage/tnfs/cloud/Dropbox/TDS.jpg");
```



# Operation for “rename”

- Move from one Cloud to another Cloud using “rename”

```
rename("/opt/storage/tnfs/cloud/Dropbox/TDS.jpg" "/opt/storage/tnfs/cloud/OneDrive/TDS.jpg");
```



# How to know the path?

- **Search our File System**

- Our File System is located in “/opt/storage/t nfs/cloud”.

/opt/storage/t nfs/cloud/Dropbox/Z1.jpg

/opt/storage/t nfs/cloud/Dropbox/TDS.jpg

/opt/storage/t nfs/cloud/OneDrive/Cloudbox.jpg

/opt/storage/t nfs/cloud/OneDrive/Video/Tizen.mp4

- **Search the Media DB**

- The Information about Cloud Contents is inserted to Media DB.  
So You can also find the Path of Cloud Contents from Media DB.



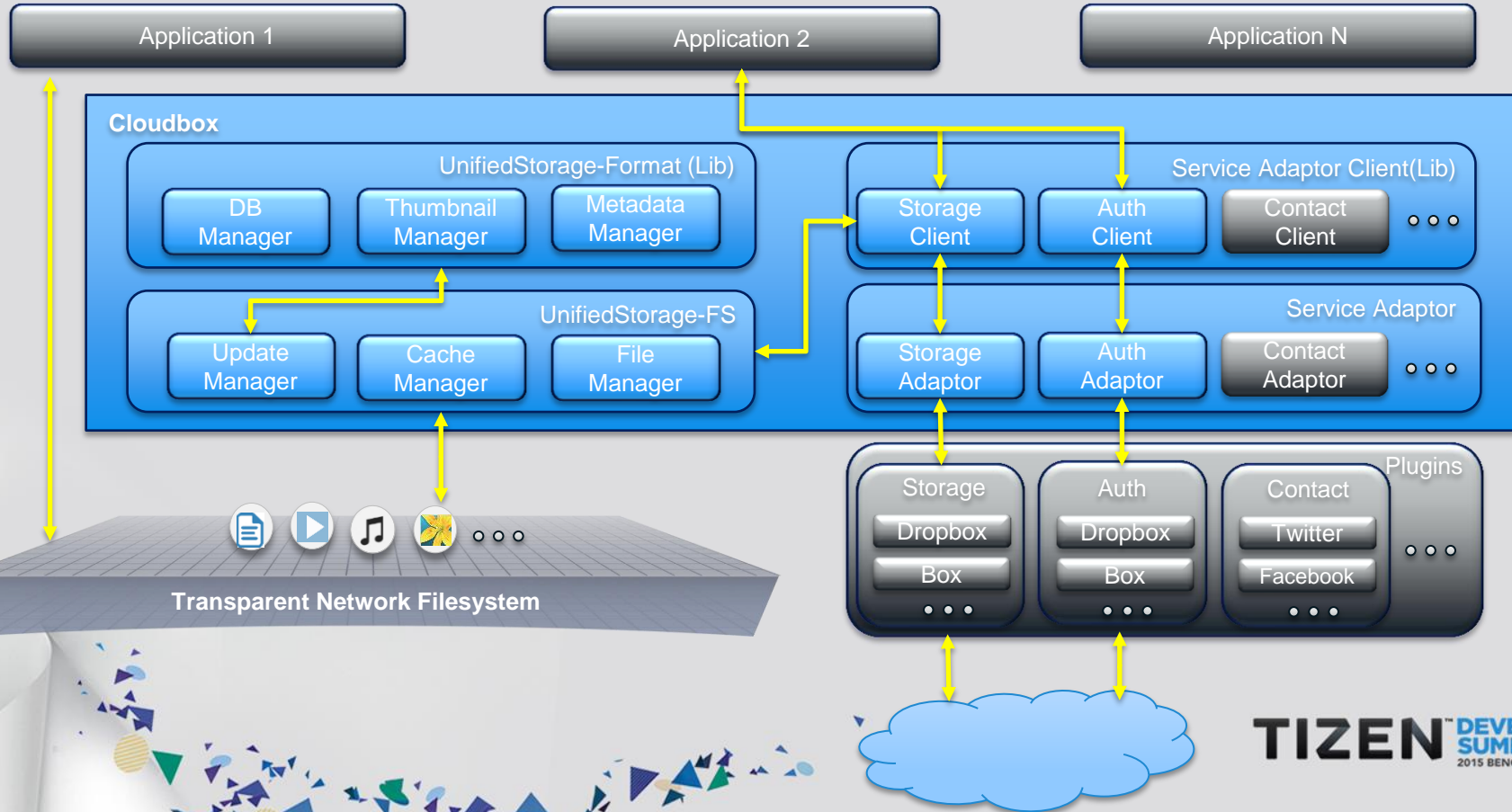
# Asynchronous for Download & Upload?

- By using below APIs, You can implement the Asynchronous feature in your Application

```
service_adaptor_result_e service_adaptor_download_file_async(service_adaptor_h handle,  
                                                             const char *server_path,  
                                                             const char *download_path,  
                                                             service_adaptor_file_h *file_handle,  
                                                             service_adaptor_error_s **error_code,  
                                                             void *user_data);
```

```
service_adaptor_result_e service_adaptor_upload_file_async(service_adaptor_h handle,  
                                                            const char *upload_path,  
                                                            const char *server_path,  
                                                            service_adaptor_file_h *file_handle,  
                                                            service_adaptor_error_s **error_code,  
                                                            void *user_data);
```

# Cloudbox Architecture



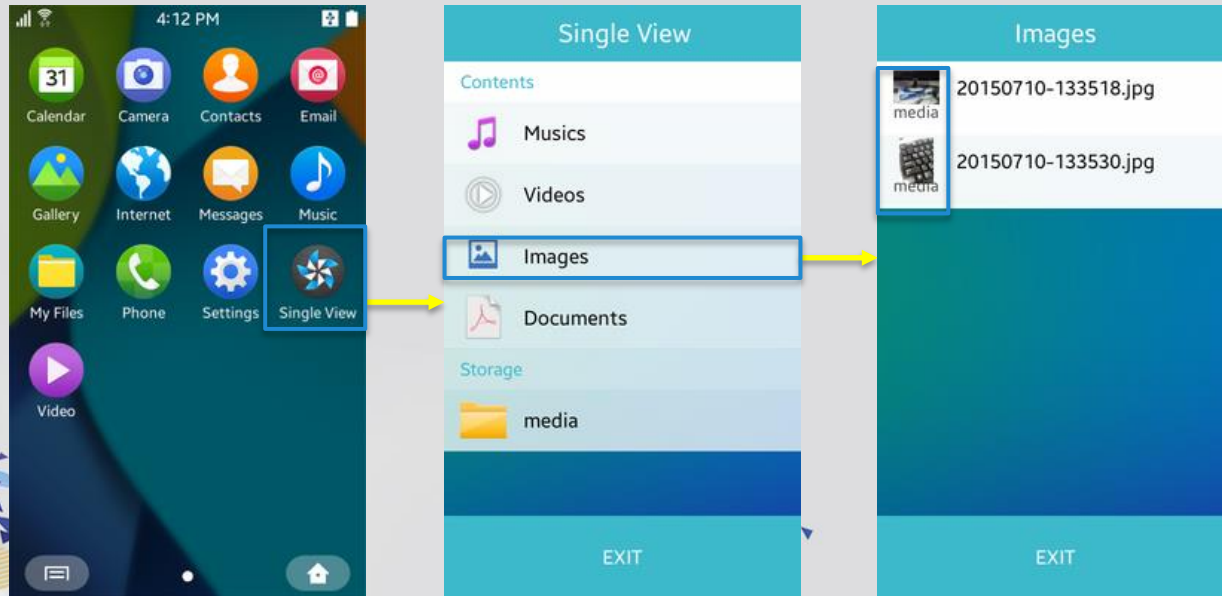
# Sample App

- Unified Single View



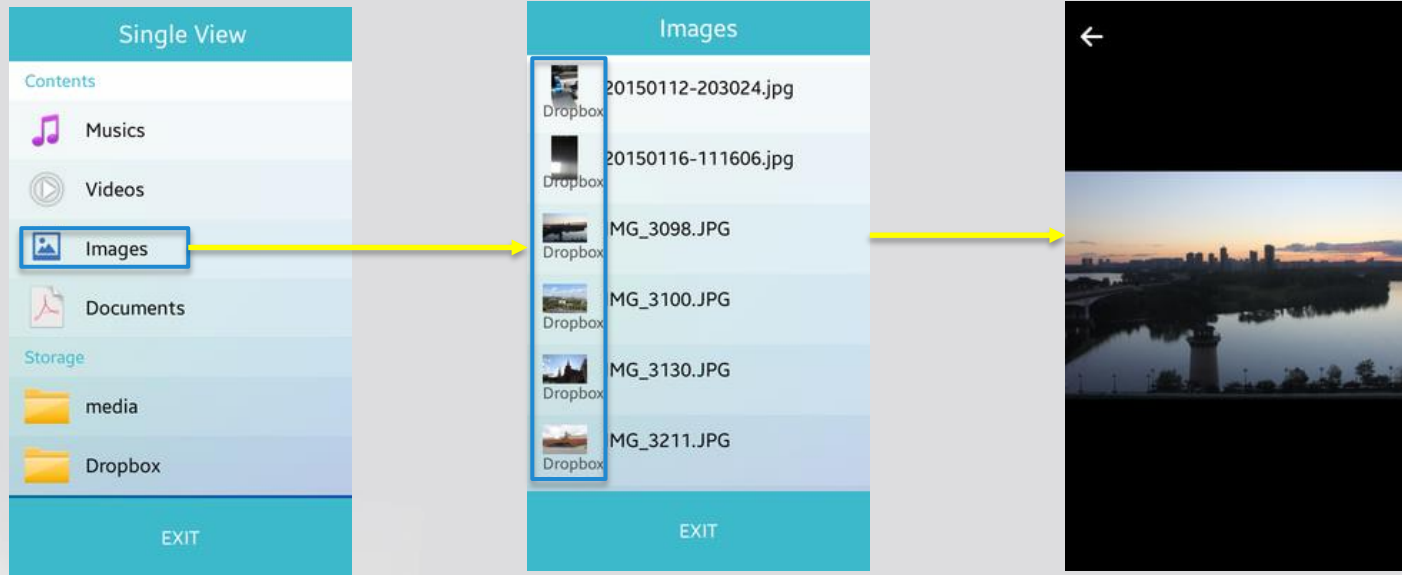
# Sample App (Unified Single View)

- In Tizen 2.4 Reference Device(Z1), Check the Cloudbox feature
- In Sample App, we just use the API only regarding Media DB
  - In Case of Local ( Not Sign in the Cloud Service)



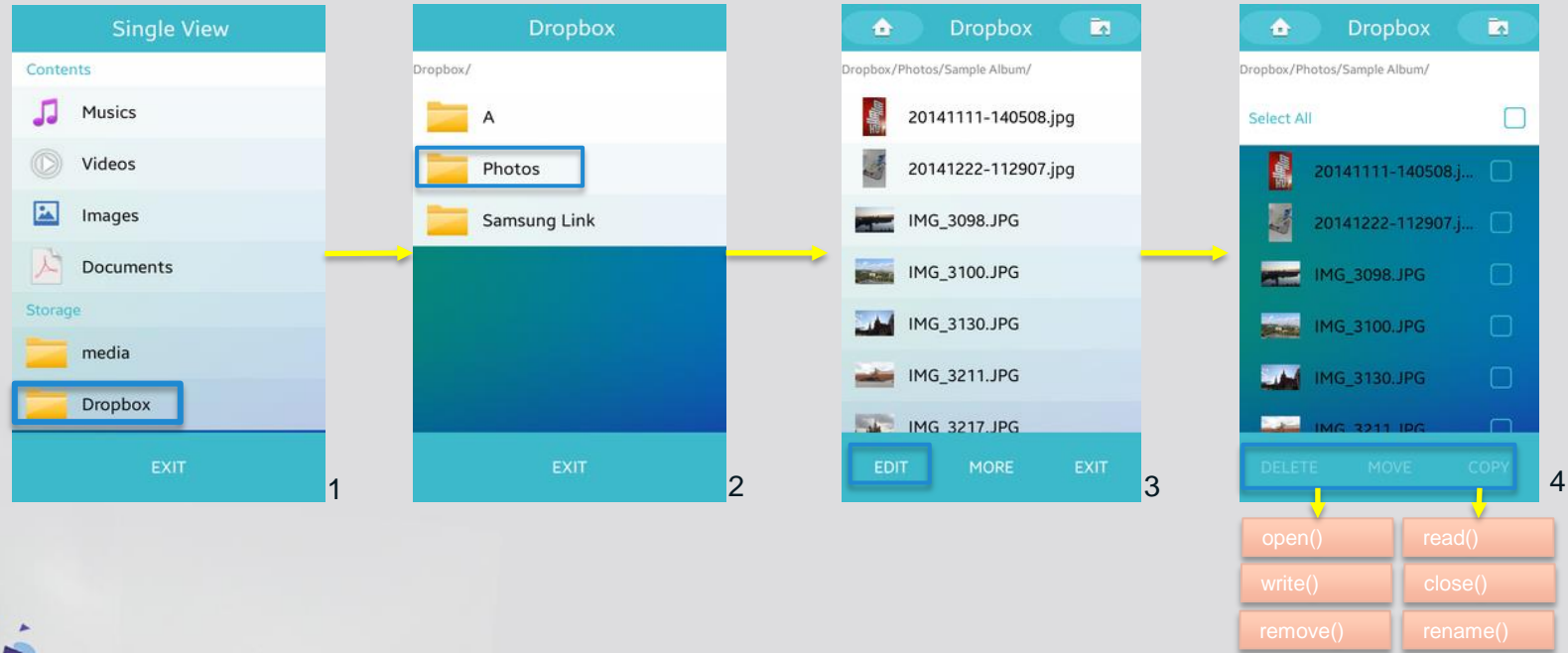
# Sample App (Unified Single View) | cont'd

- After Sign in, Cloud Contents are displayed by using `media_view` table



# Sample App (Unified Single View) | cont'd

- After Sign in, Make a Cloud Directory Easily by searching `storage` table



# Summary



# Cloudbox: Ready to go..

- **Don't worry** about accessing multi cloud contents anymore
- **Don't modify** your Application for Cloud Contents
- **Just follow 4 Check Points** only
  - APIs from Tizen 2.4 / Register / Plugins / Sign in
- **You are ready to create exciting new features** in your application using Cloudbox



## In near Future ...

- **Cloudbox is the First step in Convergence Service**
- **Extend the convergence from D2S to Device to Device**
  - Plan to Support **Unified Single View** for D2D
  - Plan to Support for **Communication** between Devices
  - Plan to **Share the Service** between Different Devices

Why don't you create new **Convergence Service** with us?

Q&A



# Thank you

**TIZEN™**  
**DEVELOPER**  
**SUMMIT**  
2015 BENGALURU   
JULY 30-31, THE RITZ-CARLTON