JACIC 一般財団法人 日本建設情報総合センター

A Feasibility Study for LDAP Certification Collaborate with Exiting Accounts in RDBMS

YOKOYAMA Yoshiyuki

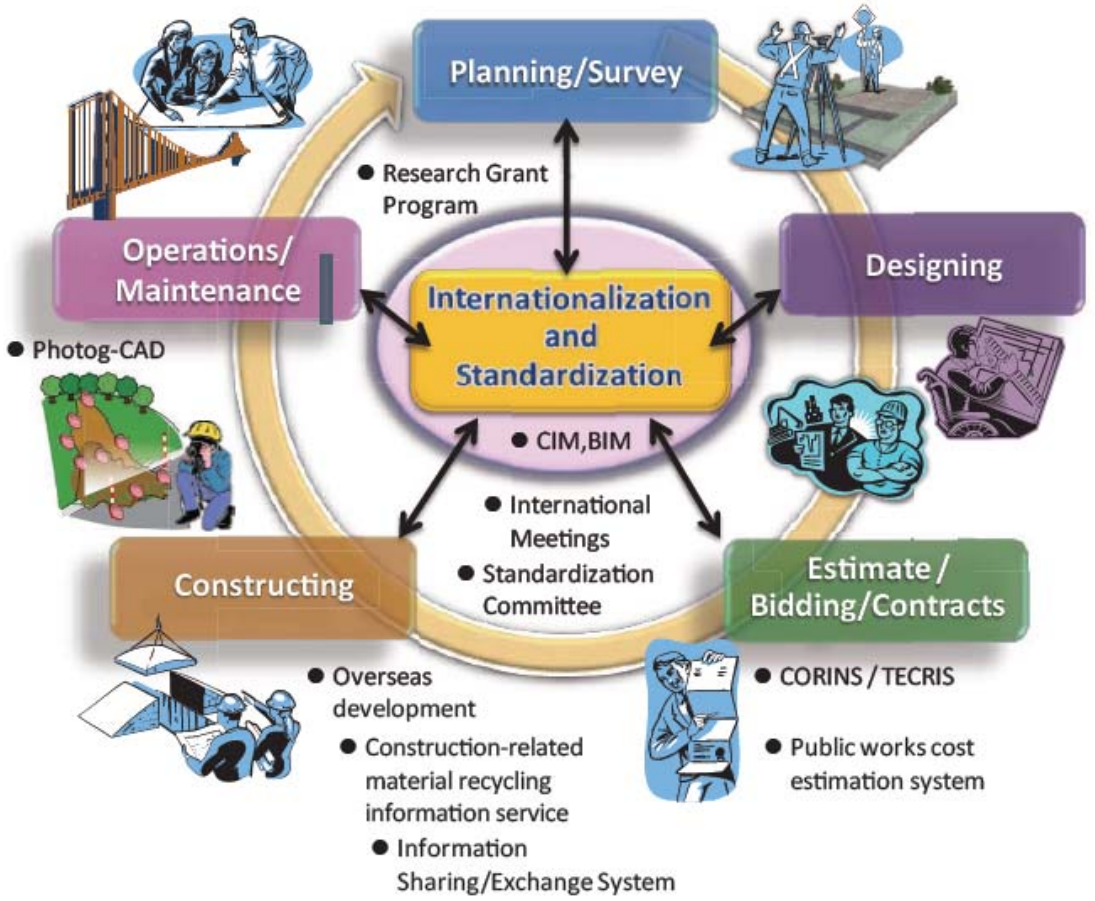
8th November ,2019

ICCBEI 2017 Sendai, Miyagi, Japan

We are JACIC

- Founded in 1985.
- Kind of non-profit organization
- Promoting ICT for construction in JAPAN.
- Aiming at whole life cycle.

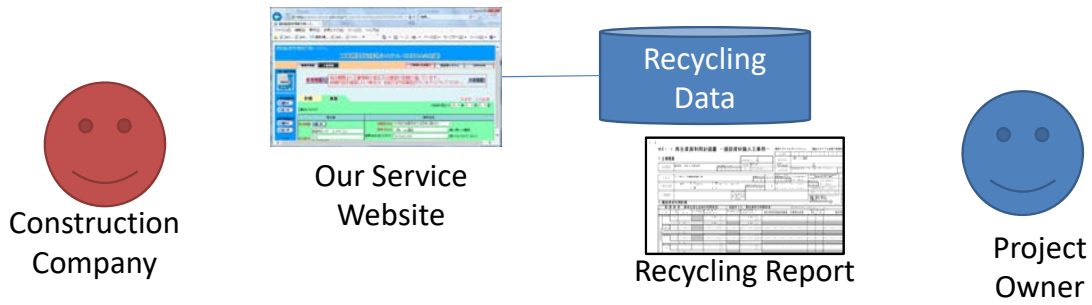
- Providing
  - Information Service
  - Research & Development
  - Standardization
  - Public relations and education



One of our information service

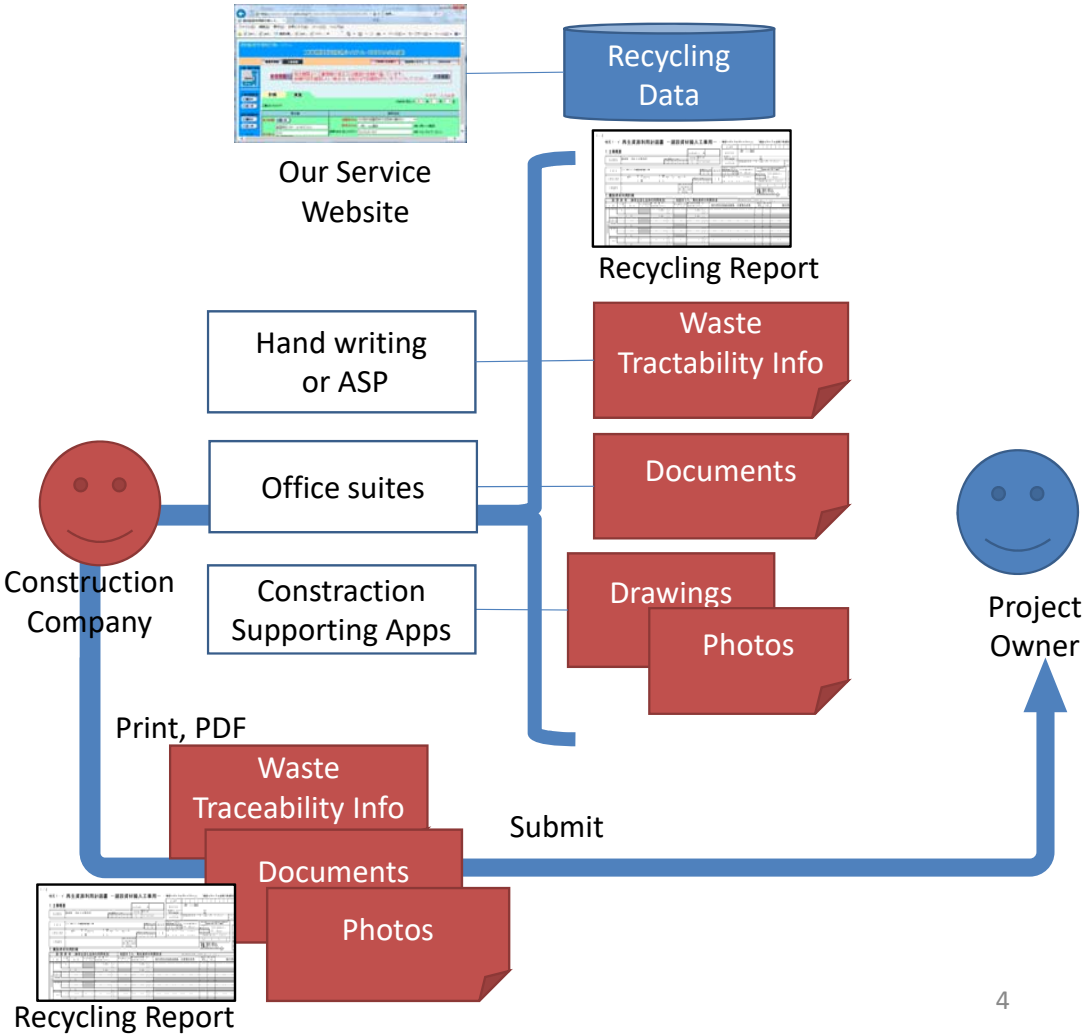
- “Construction related materials recycling information service”
- Web based information service

- This service enables our users
  - to share recycling data between project owner and construction company
  - By using recycling report form



Remaining paper oriented hand working

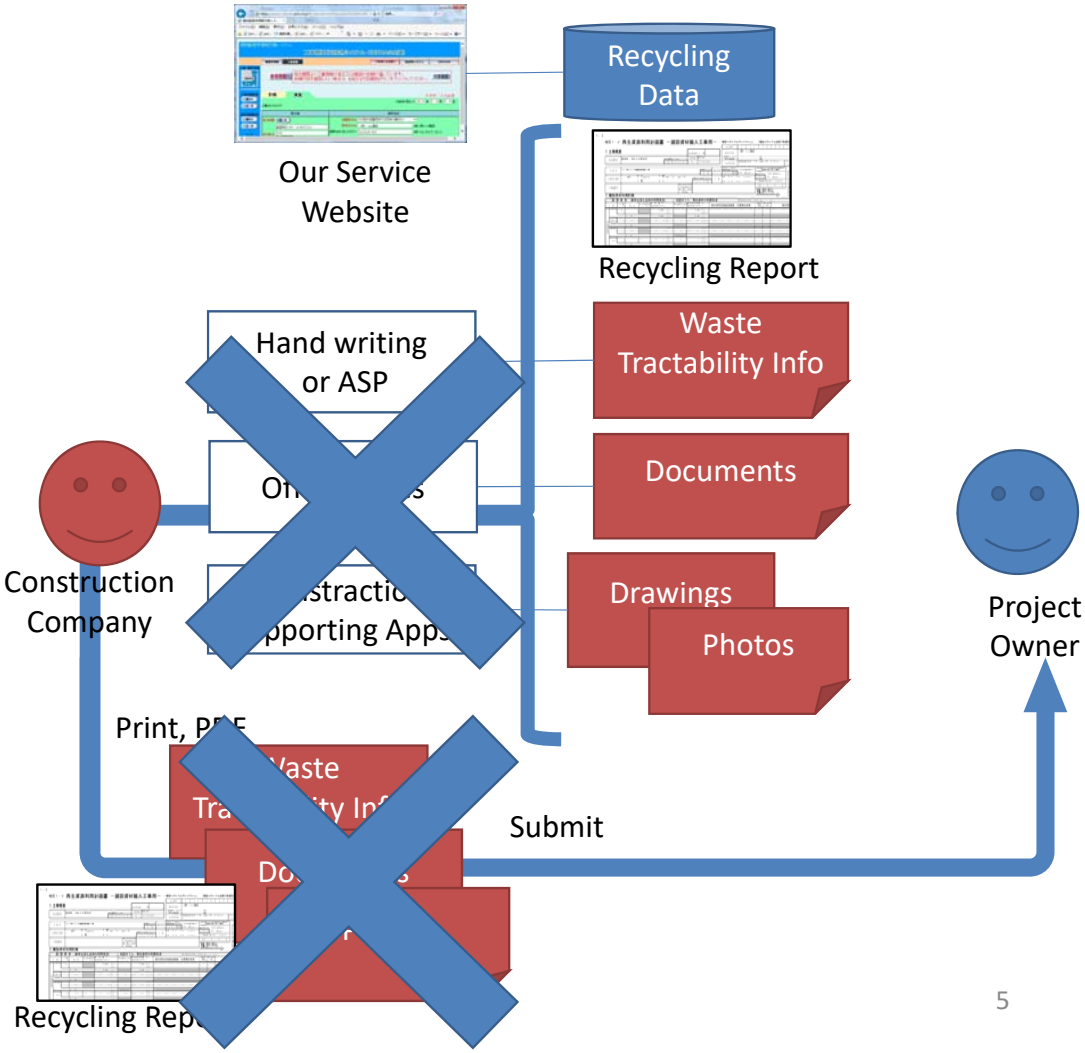
- During the construction projects, a lot of documents or data files are generated
- Construction companies must submit papers or PDF to project owner including hand-writing forms.
- Recycling Report form , printed by using our Service is also one of them.
- Then, Project owner may bind them into a folder physically for keeping.



How to avoid these burden works

- Integrating these burden process and data into Our Service seems excellent.
- However, NOT practical , because of fewer opportunity to use Our Service during construction project.

- ➢ Binding many kind of documents into the one same media ,instead of the physical folder
- ➢ Lather effective than integrating process and data.
- ➢ Short term solution, necessary until BIM will come to recycling.

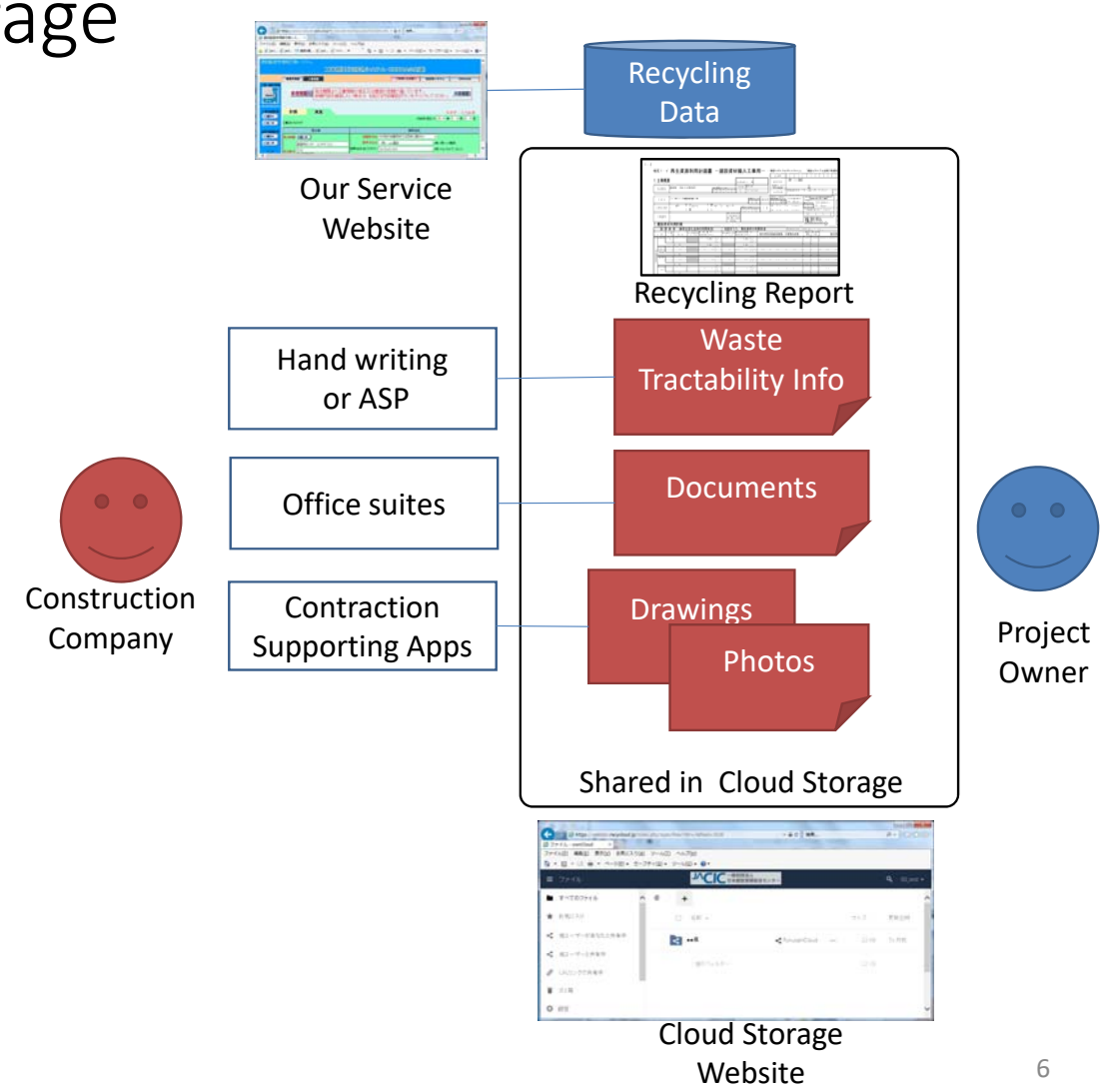


File sharing via cloud storage

- An answer for the one same media
- Can reduce user's paper oriented hand work.

- Users just have to put and brows their files via cloud storage as the one same media.
- Recycling Report and other documents are all in the one same media.

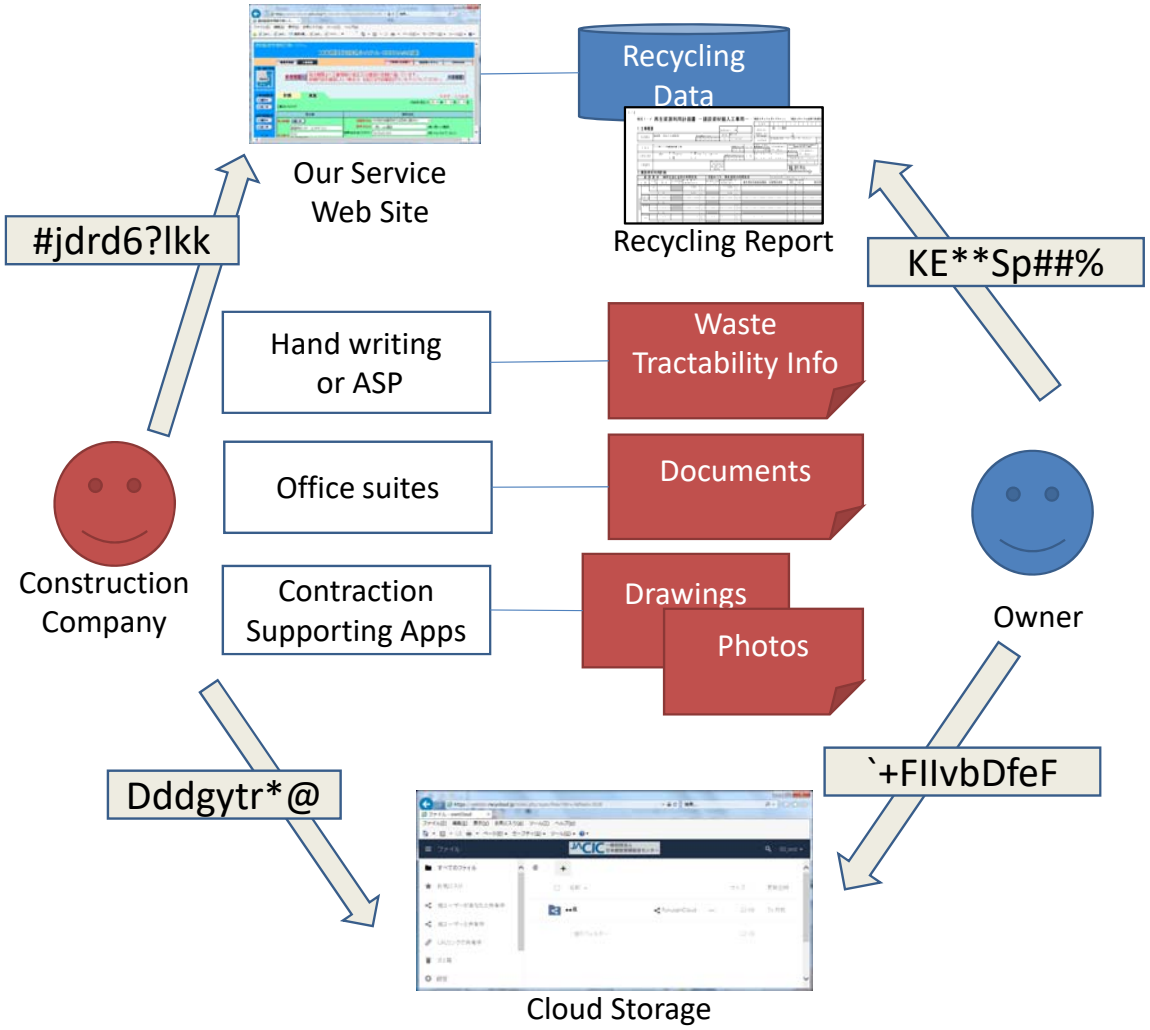
➢ Owncloud, an open source product, was chosen for our trial by its low cost .



Single Sign On is needed

- Owncloud is different service and independent from Our Service.
- It may force 50,000 user to have another new account.
- 50,000 extra admin process will be needed for registration of the new accounts.

- ➢ Single Sign On(SSO) is needed for avoiding to have multiple account.
- ➢ 50,000 users can use same account for accessing cloud storage by SSO.



Types of Single Sign On Solutions

- LDAP was chosen .
- Older but stable ,low impact for Our Service and less constraint for network.

	DB sharing	LDAP	SAML	OpenID	OpenID Connect
History	-	1993	2002	2006	2014
Methodology	Sharing user accounts in the database		Federation		
User's sign-on action	For each service or once, as long as the same domain		Once		
Protocol message	DBMS	LDAP	HTTP(s) SOAP/XML	HTTP(s) REST/XML	HTTP(s) REST/JSON
Network	Internal usage		Via Internet		
Users	Small group to Enterprise	Small group to Enterprise	Enterprise and Academic	More opened social account login	
Impact for Our Service	None	None	Necessary	Necessary	

How to certificate users

- Owncloud version9 has an add-on module to implement SSO.
- It is LDAP client, certificates uses by inquiring LDAP server.
- It helps us to reduce developing cost for a new LDAP client.

	DB sharing	LDAP	SAML	OpenID Connect
ownCloud 9.0.11	—	LDAP user and group backend 0.8.0 (Client)	—	—
Nextcloud 13.0.11	User and Group SQL Backend3.1.0 (Client)	LDAP user and group backend 1.2.1 (Client)	SSO&SAML authentication1.4.2 (Server and Client)	Social Login 1.13.0 (Client)

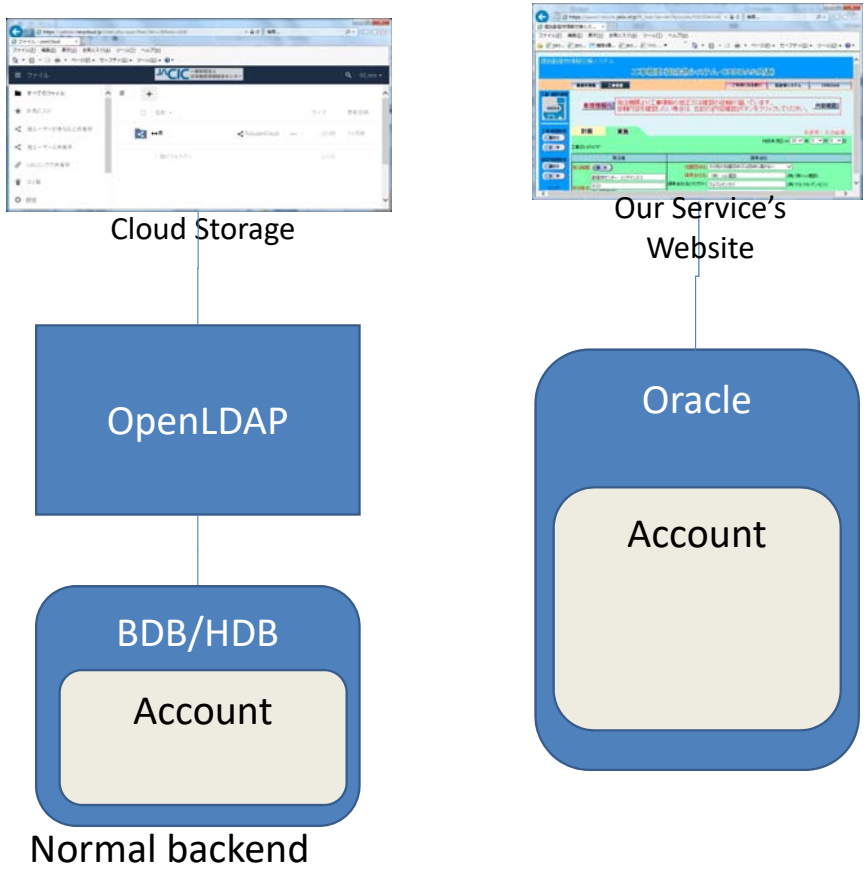
Basis of OpenLDAP

- OpenLDAP is also an open source products, implements LDAP server.
- Already installed into CentOS6.9, one of the major Linux distributions, we'd already used.
- Naturally OpenLDAP was chosen for LDAP server.

Normal backend structure of OpenLDAP

- In normal use of LDAP,BerkleyDB or HierarchicalDB is used for backend of user account

- However, No connection route to Our Service and its account in Oracle database.

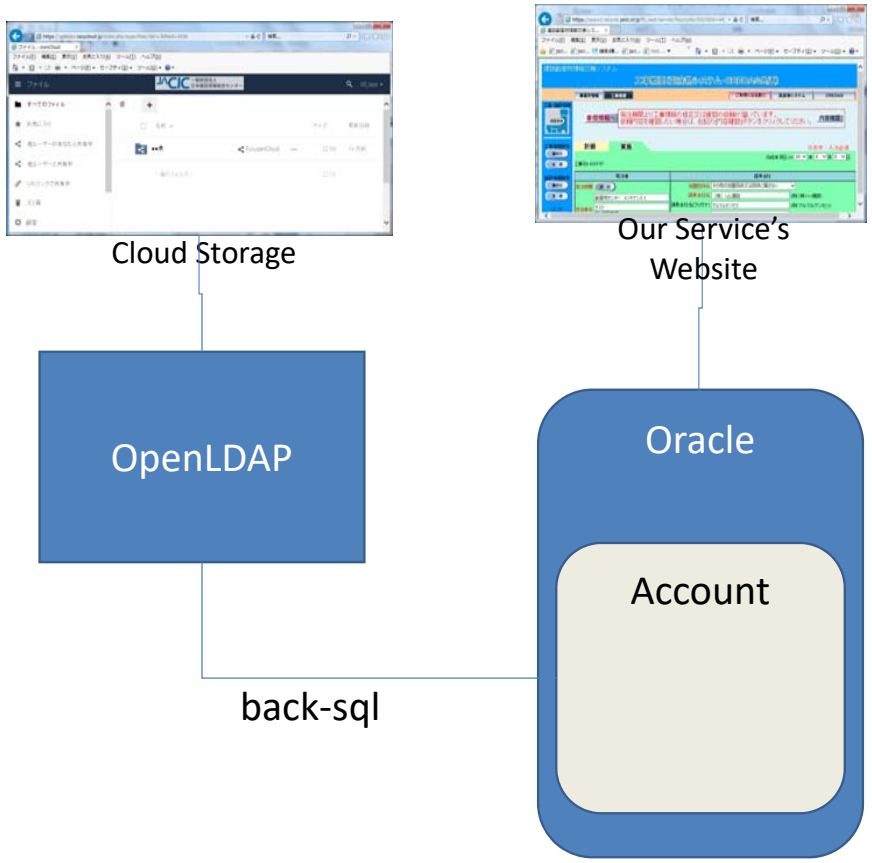


How to connect with existing account in RDBS(Oracle)

- OpenLDAP can use SQL based database, generally RDBMS, for its backend,so-called back-sql.
- Some of RDBMS ,including Oracle, is available via back-sql

- Now, existing 50,000 user accounts will be used
- Without any admin process, tools and any programing

- We taught.

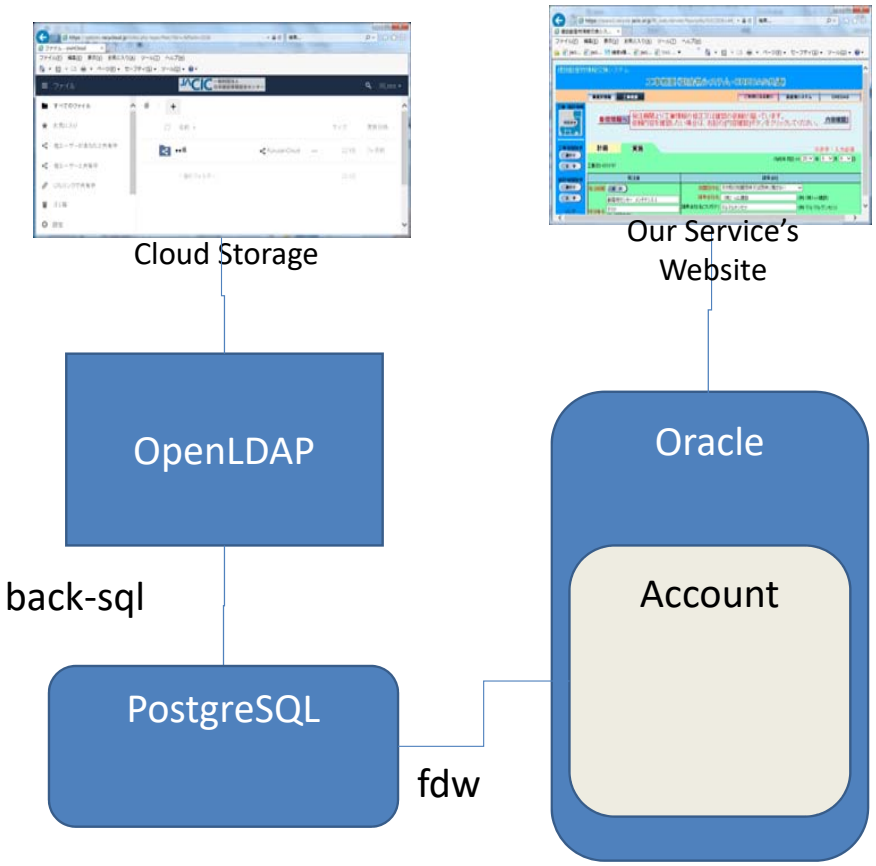


Obstacles related to product dependency

	Obstacles	Solution
PHP For LDAP client	Owncloud's add-on module for LDAP can't call ldap related function. PHP 5.6.31 in CentOS6.9 wasn't compiled with LDAP module.	Changed PHP 5.6.25 package which was compiled with ldap module From Red Hat Software Collections (RHSCL) repository.
OpenLDAP For LDAP server	`ldapsearch` ended with SQL error on Oracle. OpenLDAP2.4.xx access properly for only PostgreSQL. Generated SQL SELECT statement including text() function which is depending on PostgreSQL.	Ldap>>PostgreSQL>>Oracle
PostgreSQL For LDAP backend	PostgreSQL under ver8 can't access external database.	Upgrading PostgreSQL ver9. fdw (forgin data wrapper) plug in can access to Oracle Database.

Final backend structure

- Owncloud and Our Service are for frontend.
- Using Owncloud's SSO add-on module as LDAP client in Oncloud.
- OpenLDAP is for LDAP server.
- PostgreSQL was only available for back-sql.
- Enable to use existing Accounts in Oracle Database via fdw(forgin data wrapper) of PostgreSQL



7 Tables needed for expressing LDAP schema

domain	
id	name
1	example

units	
id	name
1	people

persons			
id	uid	name	password
1	sample1	Sample User	sample

ldap_entry_objclasses	
entry_id	oc_name
1	dcObject
2	organization
3	organizationalUnit

ldap_entries					
id	dn	dn_ru	oc_map_id	Parent	keyval
1	dc=example.dc=com	MOC=CD,ELPMAXE=CD	1	0	1
2	ou=people,dc=example,dc=com	MOC=CD,ELPMAXE=CD,ELPOEP=UO	2	1	1
3	uid=sample1,ou=people,dc=example,dc=com	MOC=CD,ELPMAXE=CD,ELPOEP=UO,1ELPMAS=DIU	3	2	1

ldap_oc_mappings						
id	name	keytbl	keycol	create_proc	delete_proc	xpect_return
1	organization	domains	id			0
2	organizationalUnit	units	id			0
3	inetOrgPerson	persons	id			0

ldap_attr_mappings										
id	oc_map_id	name	sel_expr	sel_expr_u	from_tbls	join_where	add_proc	delete_proc	paam_order	expect_return
1	1	dc	name		domains				3	0
2	1	o	name		domains				3	0
3	2	ou	name		units				3	0
4	3	uid	uid		persons				3	0
5	3	cn	name		persons				3	0
6	3	sn	name		persons				3	0
7	3	userPassword	password		persons				3	0

2 tables have to be dynamically generated

domain	
id	name
1	example

units	
id	name
1	people

persons			
id	uid	name	password
1	sample1	Sample User	sample

ldap_entry_objclasses	
entry_id	oc_name
1	dcObject
2	organization
3	organizationalUnit

ldap_entries					
id	dn	dn_ru	oc_map_id	Parent	keyval
1	dc=example.dc=com	MOC=CD,ELPMAXE=CD	1	0	1
2	ou=people,dc=example,dc=com	MOC=CD,ELPMAXE=CD,ELPOEP=UO	2	1	1
3	uid=sample1,ou=people,dc=example,dc=com	MOC=CD,ELPMAXE=CD,ELPOEP=UO,1ELPMAS=DIU	3	2	1

ldap_oc_mappings						
id	name	keytbl	keycol	create_proc	delete_proc	xpect_return
1	organization	domains	id			0
2	organizationalUnit	units	id			0
3	inetOrgPerson	persons	id			0

ldap_attr_mappings										
id	oc_map_id	name	sel_expr	sel_expr_u	from_tbls	join_where	add_proc	delete_proc	paam_order	expect_return
1	1	dc	name		domains				3	0
2	1	o	name		domains				3	0
3	2	ou	name		units				3	0
4	3	uid	uid		persons				3	0
5	3	cn	name		persons				3	0
6	3	sn	name		persons				3	0
7	3	userPassword	password		persons				3	0

For generating 1st table

- Created the view from original `user account` table in Oracle
- Created foreign table in postgreSQL as `persons` LDAP schema

【PostgraSQL】

```
create server oraclehostname foreign data wrapper oracle_fdw
options (dbserver '//oraclehostname:1521/ServiceName');

create user mapping for ldap server oraclehostname options(user
'ldap_test_pg', password 'passwordtext');
create foreign table persons(
        id integer,
        uid varchar(255),
        name varchar(255),
        password varchar(64))
        server oraclehostname
         options(schema 'LDAP_TEST_PG', table 'LDAP_PERSONS');

ALTER FOREIGN TABLE persons OWNER TO ldap;
```

【Oracle】

```
create view ldap_persons
as SELECT
    row_number() OVER (ORDER BY A.USER_ID) as id,
      A.USER_ID,
      A.User Display Name  ,
      '{CRYPT}'|| A.PW PW
 FROM Original user account Table A
WHERE Conditions to choose users';
```

For generating 2nd table

- Created the view in PostgreSQL as `ldap_entries` schema.
- To synchronize with users, increasing/decreasing.
- Dynamically generated tables are necessary.

【PostegrSQL】

```
create table ldap_entries_sub
 (
                id serial not null primary key,
                dn varchar(255) not null,
                dn_ru varchar(255) not null,
                oc_map_id integer not null references ldap_oc_mappings(id),
                parent int NOT NULL,
                keyval int NOT NULL,
                UNIQUE ( oc_map_id, parent,keyval ),
                UNIQUE ( dn )
);

insert into ldap_entries_sub (id,dn,dn_ru,oc_map_id,parent,keyval)
 values (1,'dc=example,dc=com','MOC=CD,ELPMAXE=CD',1,0,1);
insert into ldap_entries_sub (id,dn,dn_ru,oc_map_id,parent,keyval)
 values (2,'ou=people,dc=example,dc=com','MOC=CD,ELPMAXE=CD,ELPOEP=UO',2,1,1);

create view ldap_entries
as select * from  ldap_entries_sub  union SELECT
      abs(3) as id,
      text('uid=')|| A.uid ||text(',ou=people,dc=example,dc=com') as dn,
      text('MOC=CD,ELPMAXE=CD,ELPOEP=UO,')|| reverse(A.uid)||text('=DIU') as dn_ru,
   abs(3) as oc_map_id,
   abs(2) as parent,
   row_number() OVER (ORDER BY A.uid) as keyval
  FROM persons A;
```

Conclusion

- LDAP Certification Collaborate with Exiting Accounts in RDBMS is feasible.
- Found and solved version dependency for each product and combination of products.

- Provided to understand for expressing LDAP schema in RDMS.
- 7 tables and their relationships are illustrated.

- Targeted Our Service in this case study has 50,000 users.
- Another Our Service ,CORINS/TECRIS ,have more than 160,000 users in Oracle. This case study will be helpful as they adopt SSO.

Future work

- Nextcloud spin-out product from Owncloud has another SSO client.
- Easy to use much modern technologies for SSO.

	DB sharing	LDAP	SAML	OpenID Connect
ownCloud 9.0.11	—	LDAP user and group backend 0.8.0 (Client)	—	—
Nextcloud 13.0.11	User and Group SQL Backend3.1.0 (Client)	LDAP user and group backend 1.2.1 (Client)	SSO&SAML authentication1.4.2 (Server and Client)	Social Login 1.13.0 (Client)

Thank you