

Qt and Cloud Services

Sami Makkonen

Qt R&D

Digia

Different types of Cloud services

Qt and Cloud Services

Cloud API for Qt

Using PaaS Services from Qt application

engin.io

Using BaaS from Qt application

Different Type of Cloud Services

Qt and Cloud Services

Comments about Cloud from Qt Users

Please describe what services (e.g. file or table storage) and what service providers are you using/planning on using in your Qt cloud based solutions?

"Not sure yet, but we definitely need it."

"Not sure at the moment but we are getting signs from our industry that it is important."

"We haven't decided yet, but I can see it coming".

"It would be great to create Cloud based services, or even web services with Qt itself."

"We have our own servers set up for users to share files and store them online"

"Network storage and sync is important for us"

"Very large files on amazon servers, probably."

"all that are performant enough to support numerical analysis of big data / simulation tasks."

"Table storage is important for us. We want to synchronize all information about users activity between his devices."

"Blob storage, File access and database tables would all be useful."

"We would like to see an API that hides the details of the system from the programmer."

"It is critical that there be an option to allow for a local storage target."

"We are looking at how we would share files and databases in an internal (private) cloud."

"User profile storage so user can access user settings from anywhere."

"We are considering providing it ourself to ensure our security needs are met."

"We currently use AWS a lot - storage, computation, management, etc. Unfortunately due to the lack of a C++ API for AWS, all associated applications are relegated to Java. Having Qt support here would be fantastic. If Qt took on creating and supporting a C++ API for AWS, I'm sure they would draw in a large number of customers in my situation who would love to use C++ (for both GUI and non-GUI purposes) but just can't."

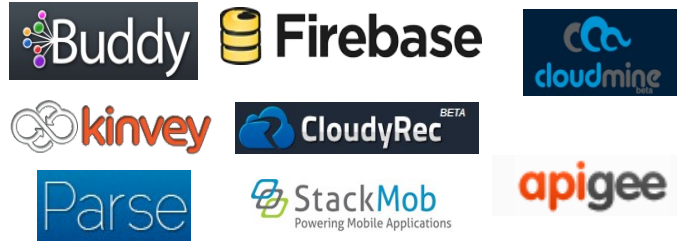
Cloud Service Models & Examples of Service Providers

SaaS
Software as a
Service



End User Applications
CRM, accounting, email, games ...

BaaS
Backend as a
Service



Cloud Storage
Functionality (e.g. user management,
push notifications, social media integration)
SDK/API

PaaS
Platform as a
Service



Computing Platform
Solution stack - Databases, Application
Servers, API

IaaS
Infrastructure as a
Service



Computing resources
Virtual Machines, Storage, Network,
Load Balancing

- Using cloud computing for intensive computations, dynamic load scaling and geographical distribution
- Hosting databases in the cloud servers and accessing them through Qt API's
- Hosting file services in the cloud and implementing a synchronization mechanism / API
- RESTful interfaces well supported
 - Good network base layer
 - XML and JSON parsing available
- SOAP based Web Service interfaces
 - Commercial implementations available e.g. from KDAB
 - Some open source implementations are also available

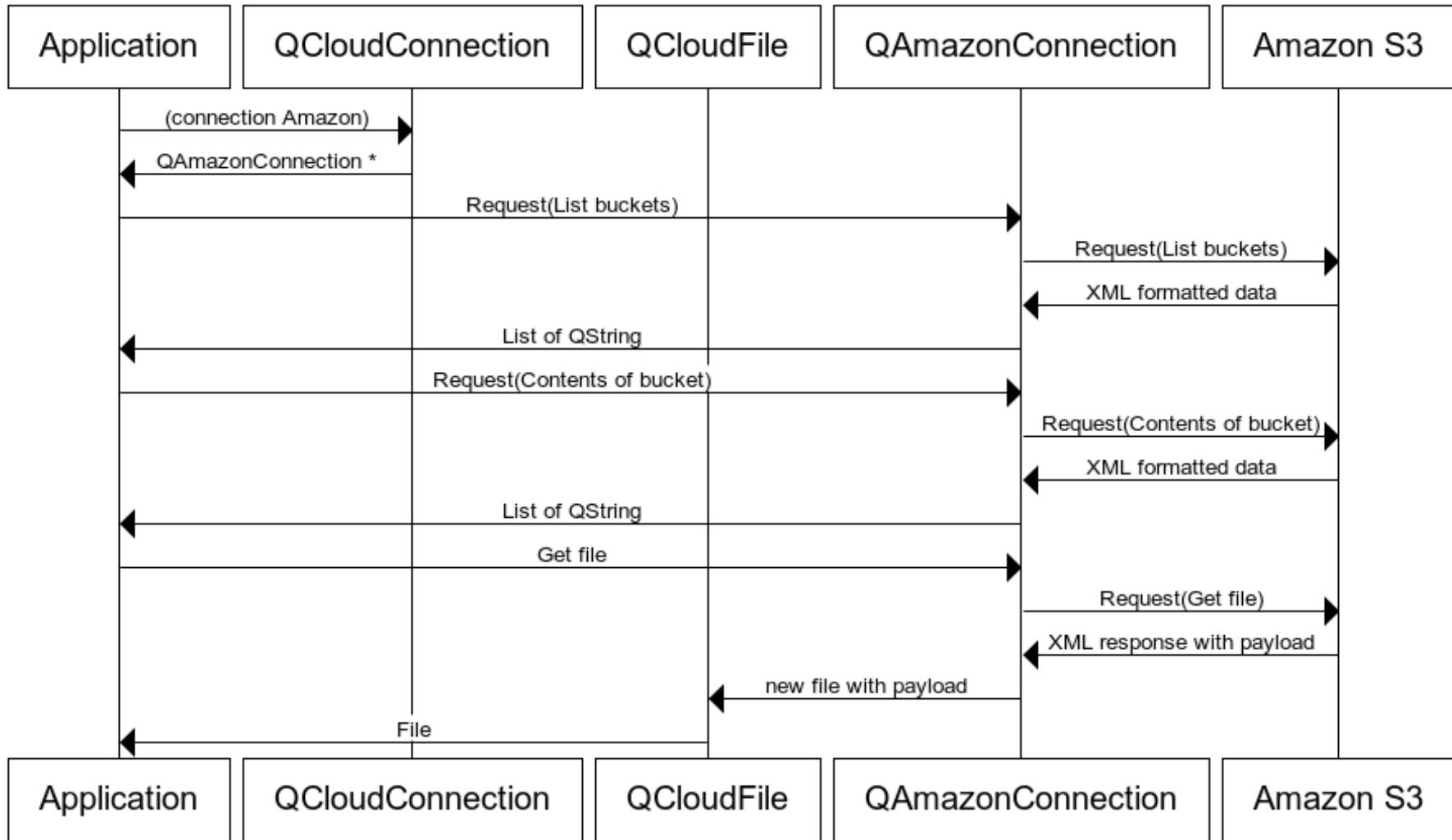


Qt Cloud API

Using PaaS from Qt

- A Research project with the University of Jyväskylä
- The idea was to investigate if we could create a harmonized Qt API for accessing different services,
- However the requirement of having one common API is not mandatory
- Amazon and Azure web services were selected as the first ones for implementation

Get certain file



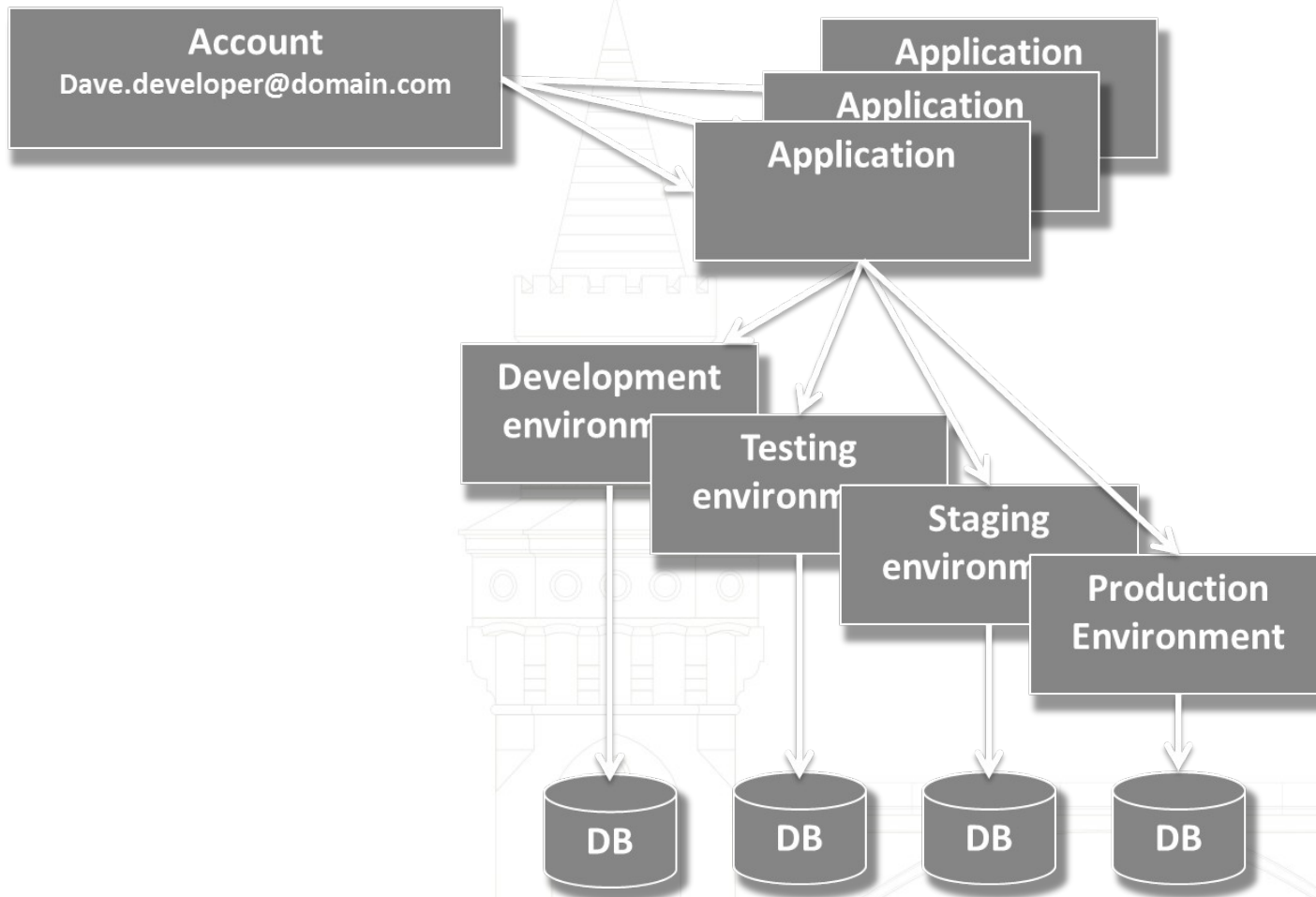
- Research prototype
 - Basic use cases for both Azure and Amazon File Storage are now working
 - One example application and several test cases
 - File storage implementation also needs work on more complicated use cases such as file/folder synchronization
 - Some preliminary implementation Table Storage exists but needs a lot of work
 - Requires Qt5 because of the SHA256 needed in authentication
- Proposed to be a Qt Project Playground project
- Developers, reviewers, testers needed – any volunteers ?

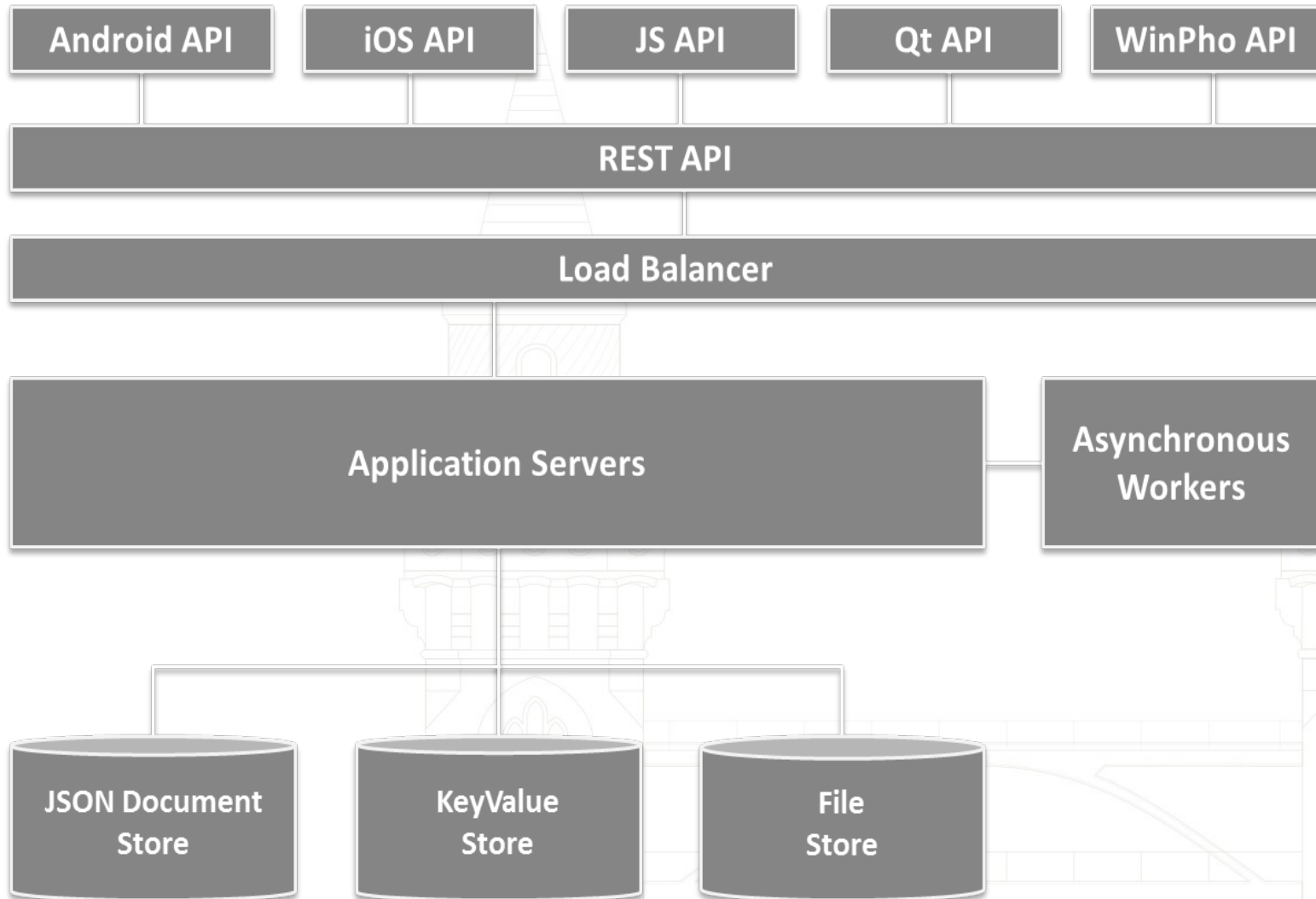


engin.io

Using BaaS from Qt

- BaaS service developed in Digia – Special focus to Qt application development
- Essentially a schema-less data storage
 - No need to define data structures in advance
 - Just store and consume JSON based data through REST APIs
 - However, engine.io can derive a schema from stored data
 - Generated schema is then used to enable various data related tasks
- engine.io provides also
 - Web based dashboard for development and management tasks
 - Manipulate & configure data when there is a need for that
 - API libraries to ease the service usage from various software platforms
 - Workflow flow support for handling application development phases
 - Development and production applications are separated versions





The screenshot shows the Qt Creator IDE with a project named 'ToDo'. The main window displays the 'main.qml' file with the following code:

```
1 /* ... */
8
9 import QtQuick 1.1
10 import "engin.io.js" as Enginio
11
12
13
```

The 'Object Schemas' dialog is open, showing a list of schemas on the left and the details for 'objects.Test1' on the right.

Name	Properties
<input checked="" type="checkbox"/> objects.Test1	3
<input type="checkbox"/> objects.Test2	0
<input type="checkbox"/> usergroups	0
<input type="checkbox"/> users	0

The 'objects.Test1' object schema details are shown in a table:

Property	Datatype	Type	Indexed
id	objectid	Internal	
createdAt	time	Internal	
creator	ref (users)	Internal	
objectType	string	Internal	
updatedAt	time	Internal	
updater	ref (users)	Internal	
done	boolean	custom	<input type="checkbox"/> x
shared	boolean	custom	<input type="checkbox"/> x
task	string	custom	<input type="checkbox"/> x

The dialog also includes a description: 'Properties defines object data structure. Enginio.io will validate json datatypes when object is created or updated. Indexing enables fulltext-search and provides faster queries but they slightly slow down insert/update operations.'

At the bottom of the IDE, the following JavaScript code is visible:

```
50
51
52 Keys.onReturnPressed: {
53     var msg = {'action': 'create', 'text': newItem.text, 'model': model};
54     updateWorker.sendMessage(msg);
55     //model.append({"task": newItem.text, "done": false})
56     //Enginio.create({"task": "\" + newItem.text + "\", "done": false})
57
58 }
```

- Server side implementation is in good shape and we are moving towards public beta phase.
- Of course the beta does not contain all the planned functionality
- Timeframe for the beta is not set yet but can express your interest by registering your email at www.engin.io
- iOS API implementation is nearly complete and example applications exist, couple of HTML/JavaScript Examples using backbone.js and JQuery
- Qt Creator plugin is an integrated WebView launching the Dashboard and most likely this the most feasible way also in the future
- Development of the Qt but needs work in development and testing. We also need more example applications



Wrap-up

- Qt is ready for the cloud in many ways
 - Use the existing applications and utilize cloud computing resources
 - Use databases and/or file storage in the cloud
 - Interface cloud servers through Web Service Interfaces
- However there are areas of improvement
 - Easy to use Qt like API's for using the Cloud services
 - Some basic standards are missing (such as oAuth)
 - Tooling support, examples
- Buzz in the BaaS Services
 - QML and BaaS services play together nicely and provide a convenient way to store data to the cloud

