Automated Testing in Service Oriented Environments

Volker Krause

KDAB
Buzzword Bingo?! 

What is this really about?
What got me into this?

AKONADI

KOLAB
Akonadi
Essentially an IMAP server
Aren't unit tests enough?

Cover all aspects of the interaction

Different implementations of the same service
Requirement

Isolation
Problems

Single instance services
Requirement

Repeatability
Complex state in the service
Availability of remote services
Enough problems, show me some solutions!
Environment Variables

#include <qtest_kde.h>

class KeyDeletionTest : public QObject
{
    Q_OBJECT
    private slots:
        void testDeleteAllPrivateKeys()
        {
            // ...
        }
};

QTEST_KDEMAIN( KeyDeletionTest, NoGUI )
#include <qtest_kde.h>

class KeyDeletionTest : public QObject
{
    Q_OBJECT
private slots:
    void testDeleteAllPrivateKeys()
    {
        // ...
    }
};

QTEST_KDEMAIN( KeyDeletionTest, NoGUI )
Environment Variables

```c
#define QTEST_KDEMAIN(TestObject, flags) \
int main(int argc, char *argv[]) \
{
    setenv("KDEHOME", ... ); \n    setenv("XDG_DATA_HOME", ... ); \n    setenv("XDG_CONFIG_HOME", ... ); \n    ...
}
```
Isolating D-Bus

Use a separate D-Bus session bus
Akonadi Testrunner

kdepimlibs/akonadi/tests/testrunner

GSOC by Igor Trindade Oliveira
Akonadi Testrunner

Separate KDEHOME/XDG_* environment

Internal D-Bus session

Akonadi Server + Test Data

Unit Test
Fake Server

class CreateJobTest: public QObject {
    Q_OBJECT

private Q_SLOTS:

    void testCreate()
    {
        FakeServer fakeServer;
        fakeServer.start();
        KIMAP::Session session("127.0.0.1", 5989);

        fakeServer.setResponse( "OK create complete" );

        KIMAP::CreateJob *job = new KIMAP::CreateJob(...);
        QVERIFY( job->exec() );
        QCOMPARE( fakeServer.request(), "A01 CREATE ..." );
        ...
    }
}
Fake Server

Easy introspection of the communication

Injection of invalid communication
Cumbersome the service contains a complex state

Might differ from the real thing
Use a virtual machine containing one instance of a service
playground/pim/akonaditest

QEMU
Virtual Machines

Testing against the real thing

Perfect isolation

Trivial to reset
Virtual Machines

Slow

Distribution of large images

Problematic with non-free services
Future work

Make it generic
Thanks for listening!

Questions?