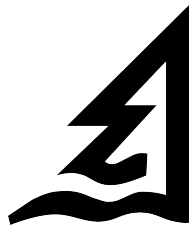


Automated Testing in Service Oriented Environments

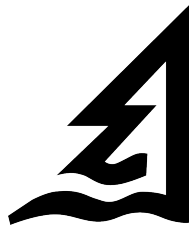
Volker Krause





Buzzword Bingo?!

What is this really about?

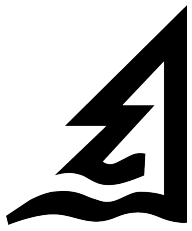


What got me into this?

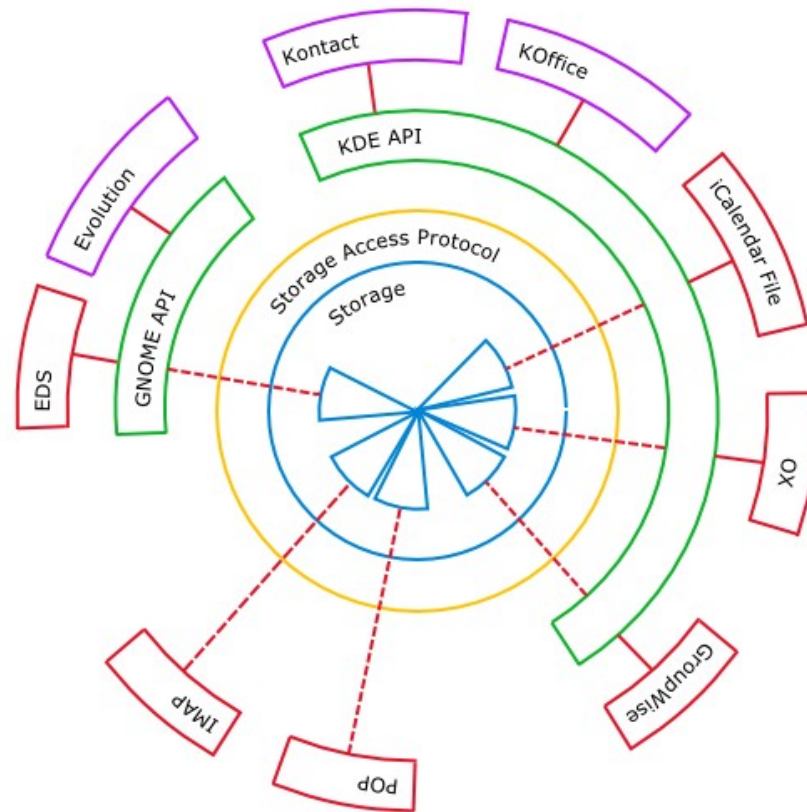


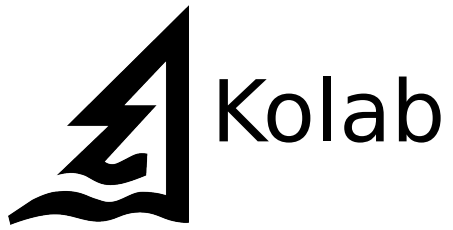
AKONADI





Akonadi



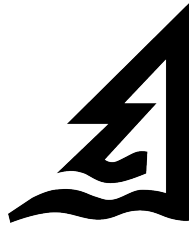


Kolab

Essentially an IMAP server

06 July 2009

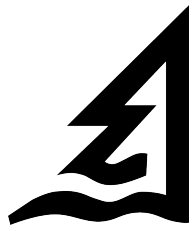
Volker Krause <vkrause@kde.org>



Aren't unit tests enough?

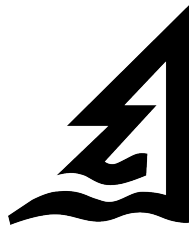
Cover all aspects of the interaction

Different implementations of the
same service



Requirement

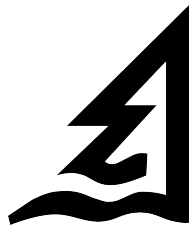
Isolation



Problems

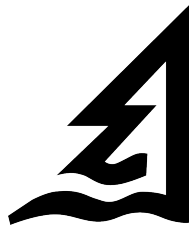
Single instance services

Repeatability



Problem

Complex state in the service

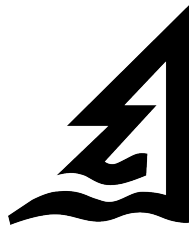


Problem

Availability of remote services

Enough problems, show me some solutions!



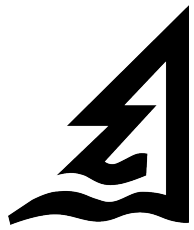


Environment Variables

```
#include <qtest_kde.h>

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

QTEST_KDEMAIN( KeyDeletionTest, NoGUI )
```

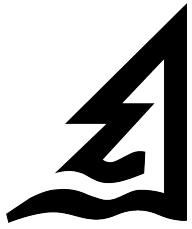


Environment Variables

```
#include <qtest_kde.h>

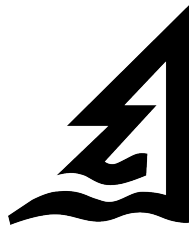
class KeyDeletionTest : public QObject
{
    Q_OBJECT
private slots:
    void testDeleteAllPrivateKeys()
    {
        // ...
    }
};

QTEST_KDEMAIN( KeyDeletionTest, NoGUI )
```



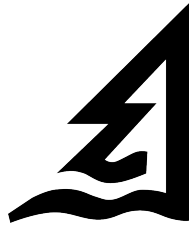
Environment Variables

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



Isolating D-Bus

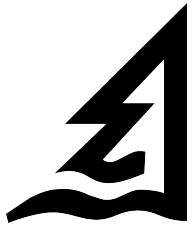
Use a separate D-Bus session bus



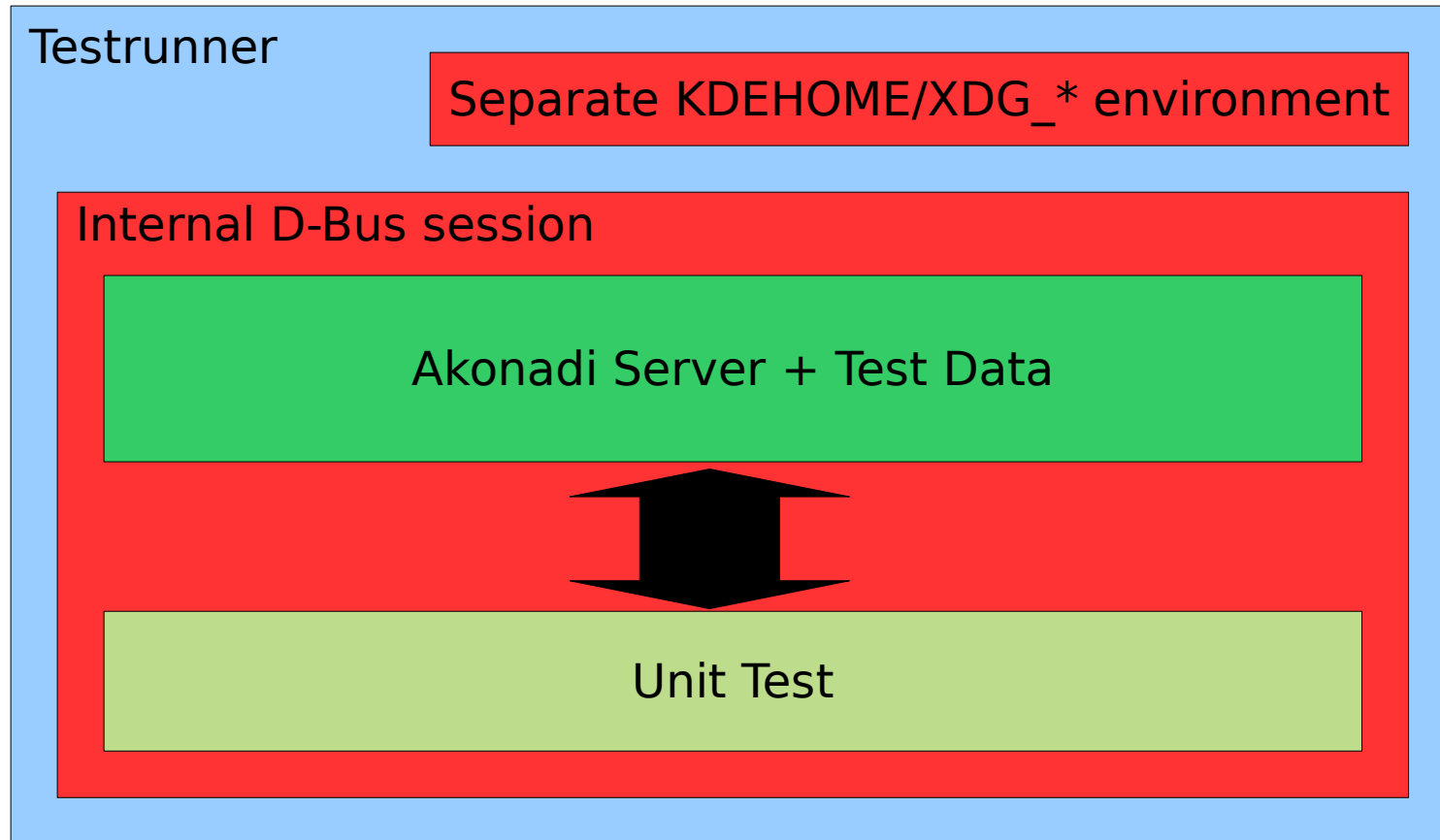
Akonadi Testrunner

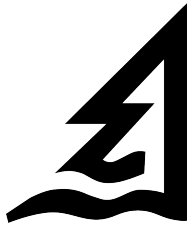
kdepimlibs/akonadi/tests/testrunner

GSOC by Igor Trindade Oliveira



Akonadi Testrunner





Fake Server

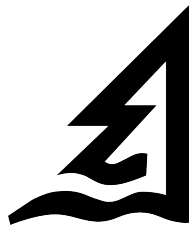
```
class CreateJobTest: public QObject {
    Q_OBJECT

private Q_SLOTS:

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

    fakeServer.setResponse( "OK create complete" );

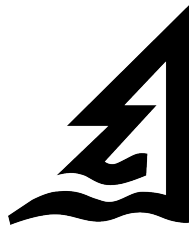
    KIMAP::CreateJob *job = new KIMAP::CreateJob(...);
    QVERIFY( job->exec() );
    QCOMPARE( fakeServer.request(), "A01 CREATE ..." );
    ...
}
```



Fake Server

Easy introspection of the
communication

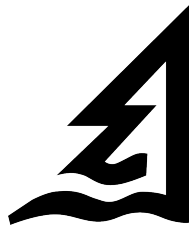
Injection of invalid communication



Fake Server

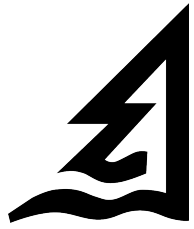
Cumbersome the service contains
a complex state

Might differ from the real thing



Virtual Machines

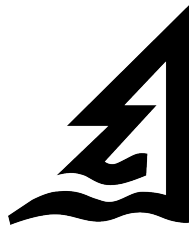
Use a virtual machine containing
one instance of a service



Akonadi Resource Tester

playground/pim/akonaditest

QEMU

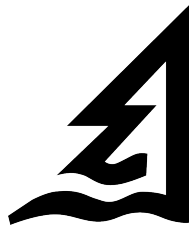


Virtual Machines

Testing against the real thing

Perfect isolation

Trivial to reset

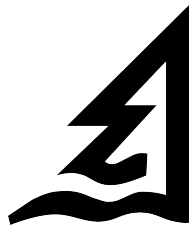


Virtual Machines

Slow

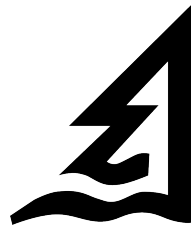
Distribution of large images

Problematic with non-free services



Future work

Make it generic



Thanks for listening!

Questions?