prolore

Implementering av Model-Based Testing i LoadRunner

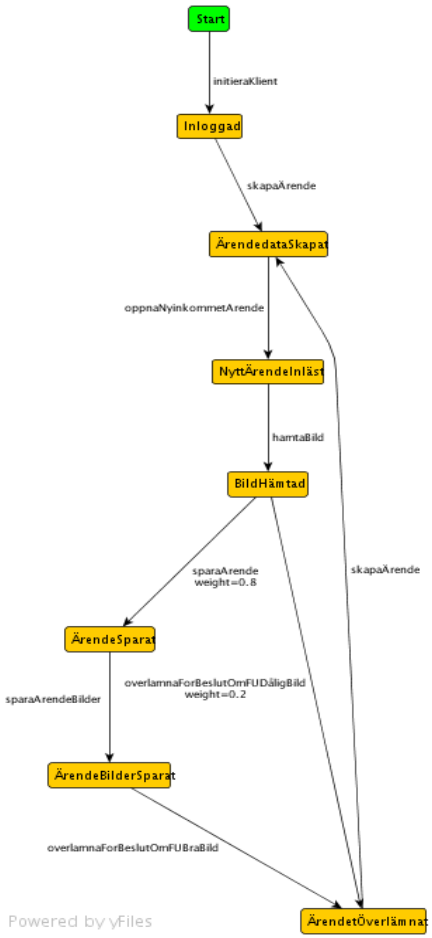
Kristian Karl

Vem står framför Er?

- Kristian Karl
- Lasttester i 7 år, testautomatiserat i 12 år
- Arbetar på Prolore (Specialiserat på Testautomatisering, Prestanda, Svarstidsmonitorering samt Utbildningar inom dessa områden)

prolore

Agenda



- Model-Based Testing
- org.tigris.mbt
- Implementationen med LoadRunner
- Sammanfattning

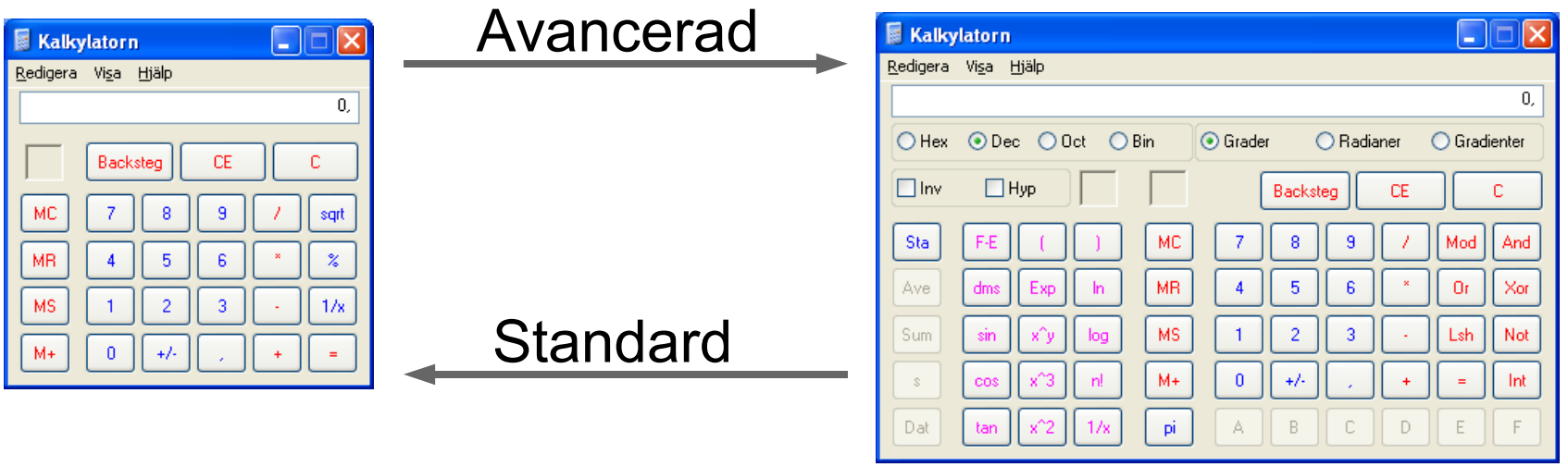
prolore

Model-Based Testing

Crash course i Model-based testing

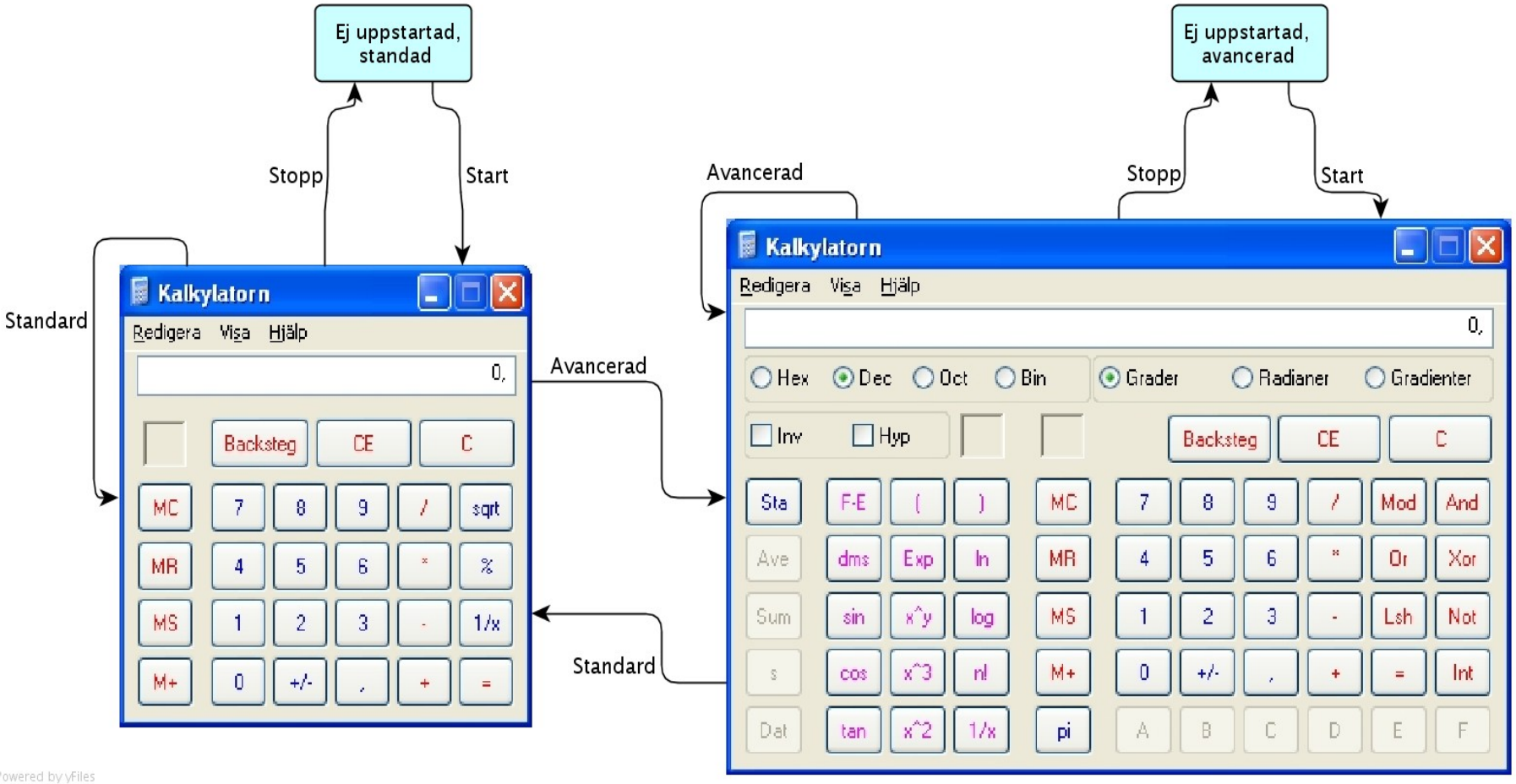
prolore

Model-Based Testing



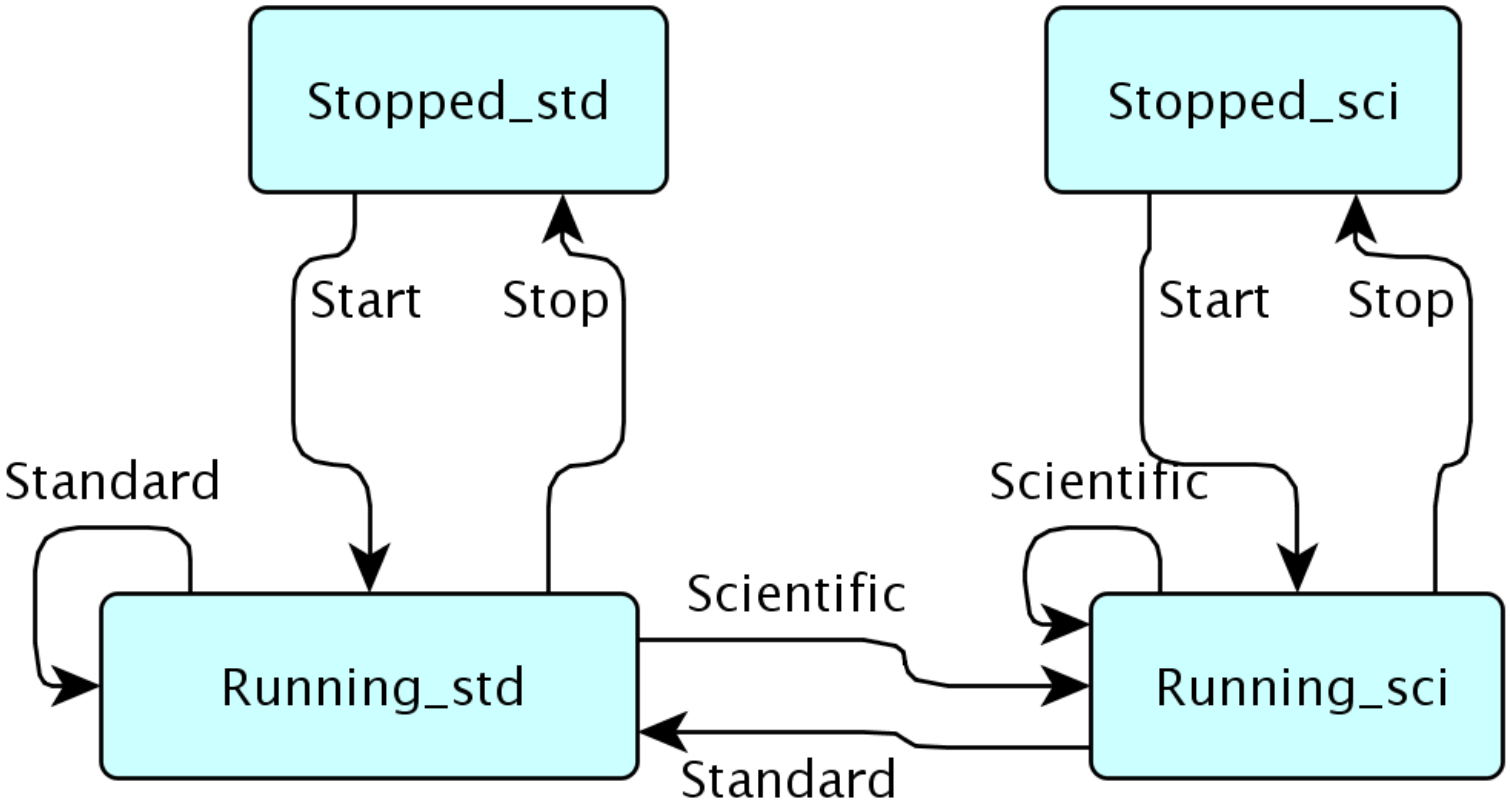
prolore

Model-Based Testing



prolore

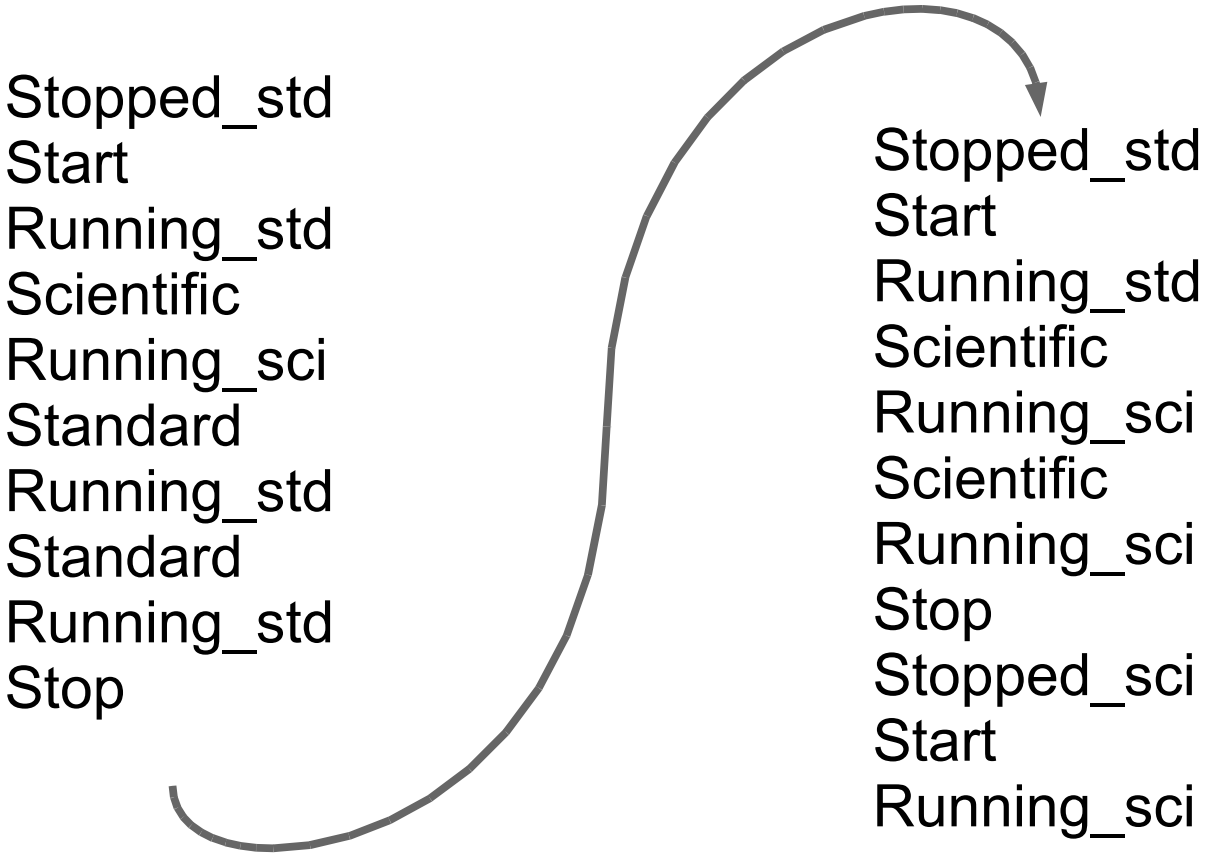
Model-Based Testing



prolore

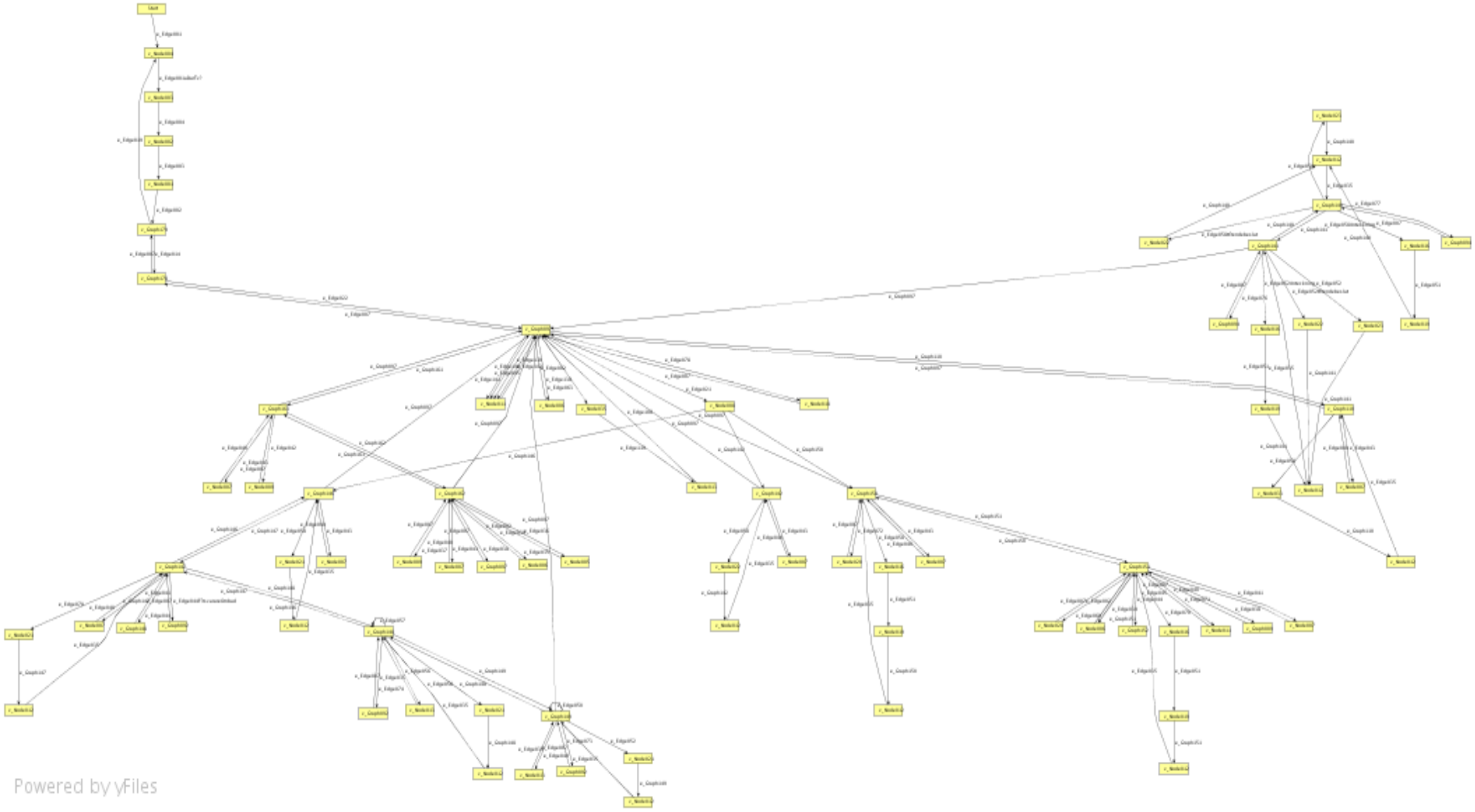
Model-Based Testing

Testfallssekvens



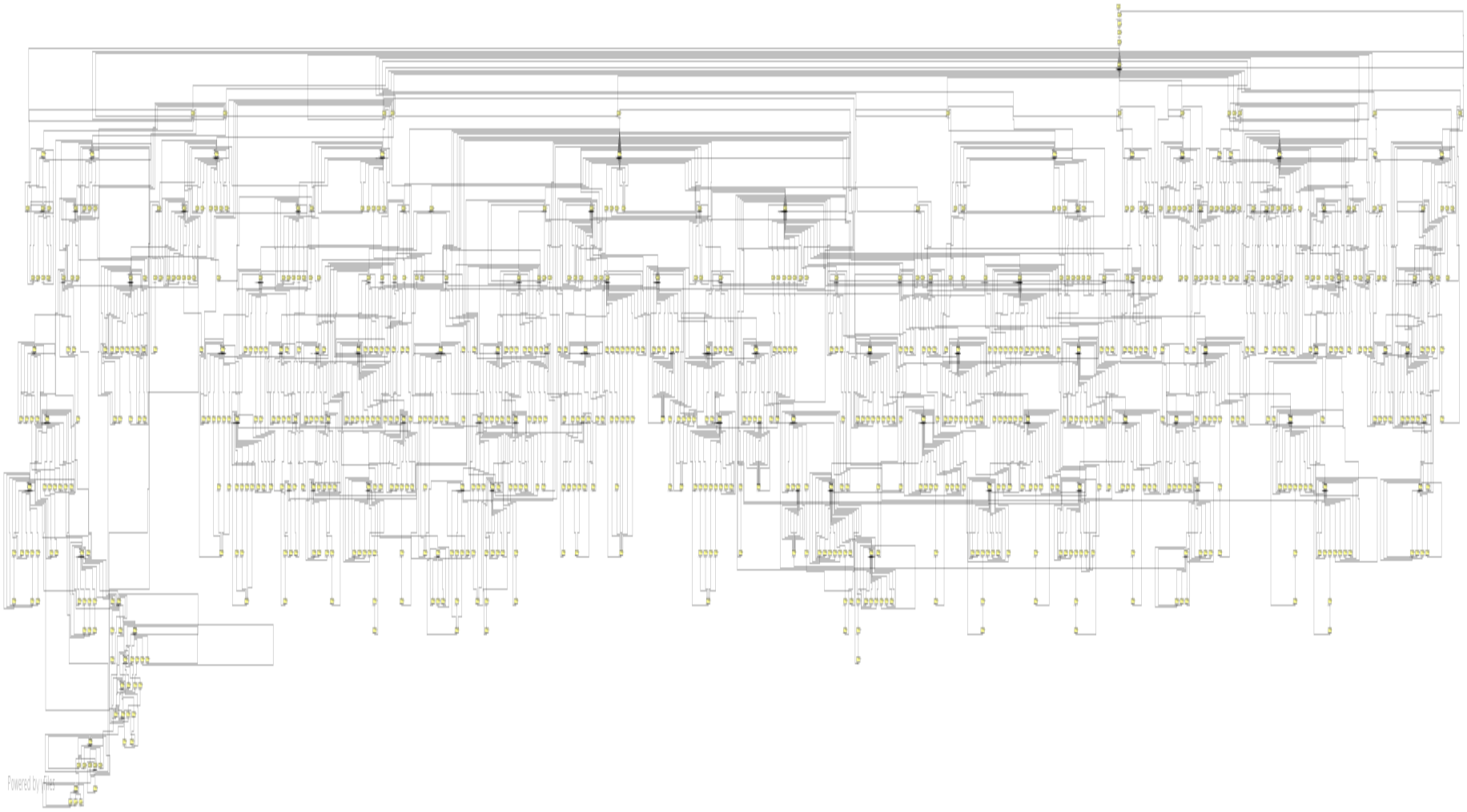
prolore

Model-Based Testing



prolore

Model-Based Testing



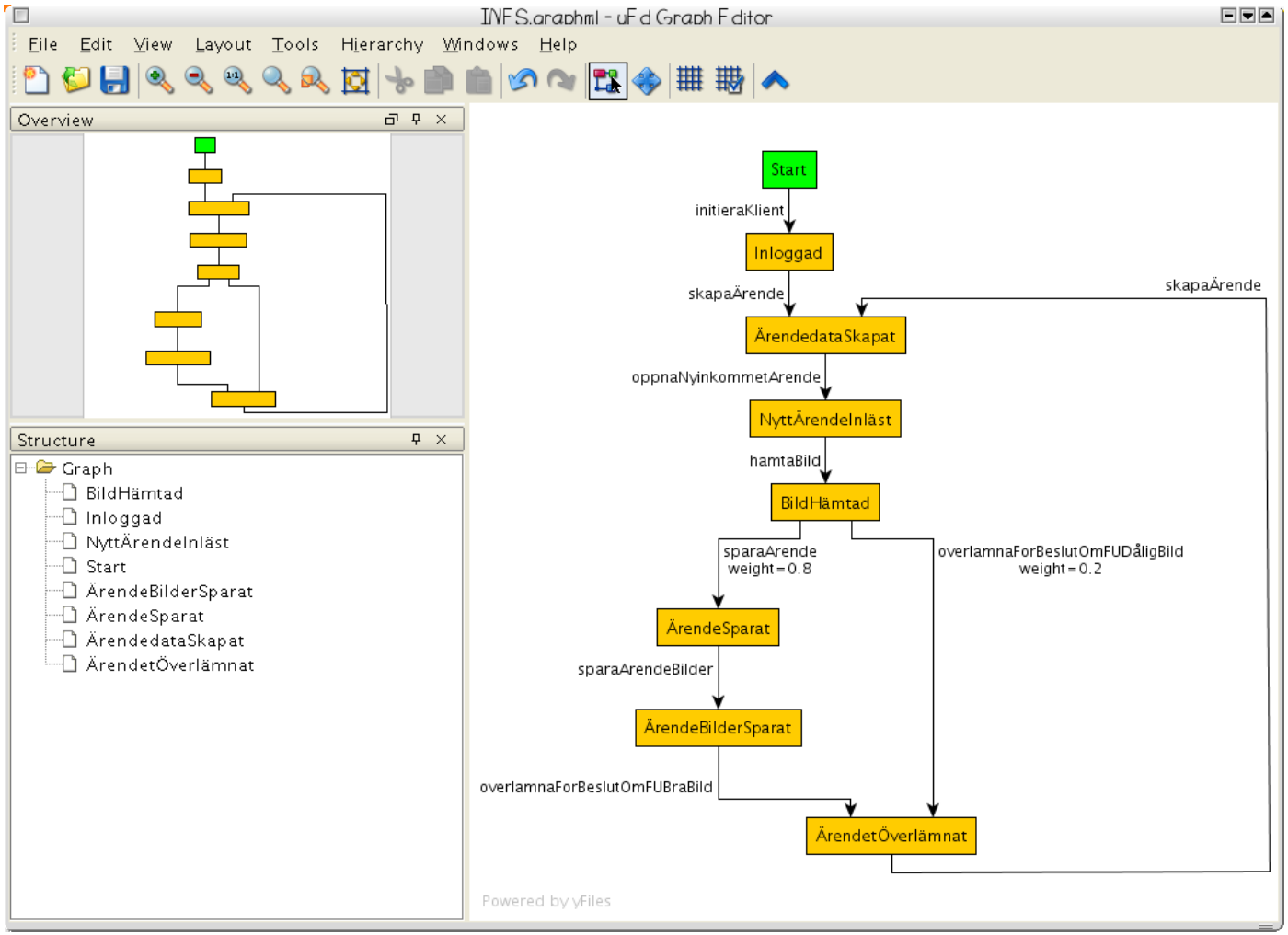
prolore

Verktyg

Verktyg för att skapa och realisera MBT

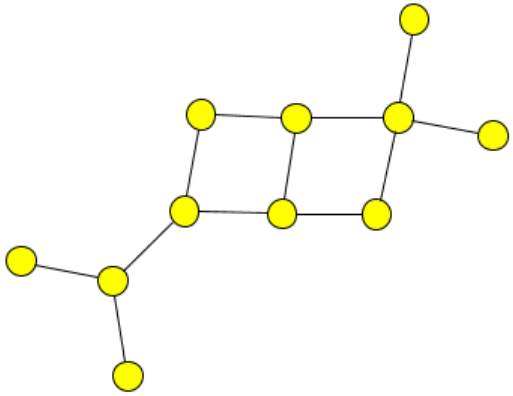
prolore

yEd (yWorks)



prolore

GraphML



- GraphML är ett filformat för grafer
- Bygger på XML
- Stödjer riktade grafer (digrafer)

prolore

org.tigris.mbt

- Klassbibliotek skrivet i Java
- Läser/skriver GraphML filer
- Genererar testfallssekvenser
- Exekverar testfallssekvenser

prolore

org.tigris.mbt

- Öppen källkod
- jung, jdom, log4j
- JUnit, Perl, LoadRunner, Functional Tester
- GUI, Web services, API

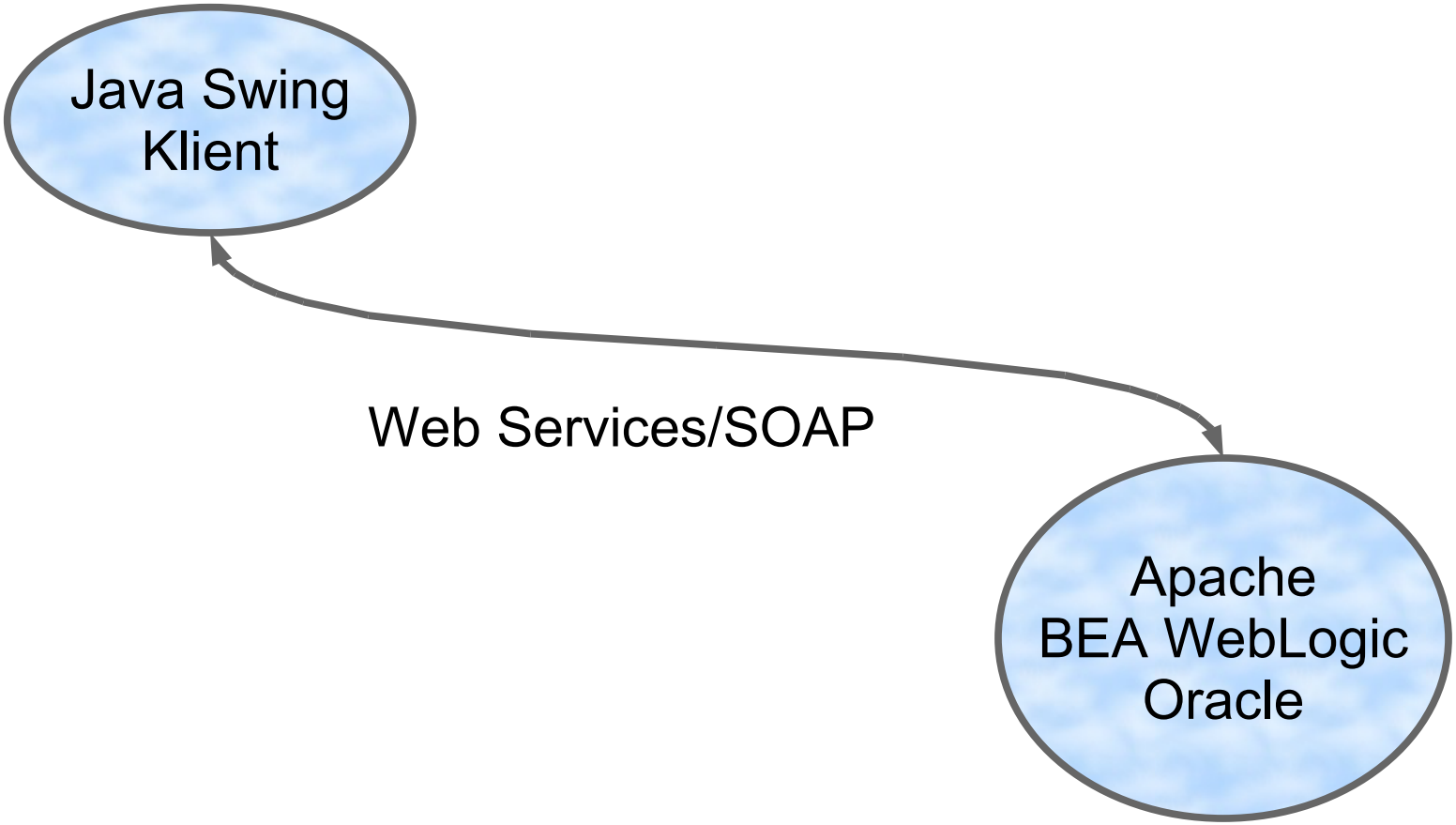
prolore

Prestandatestet

Prestandatest med MBT

prolore

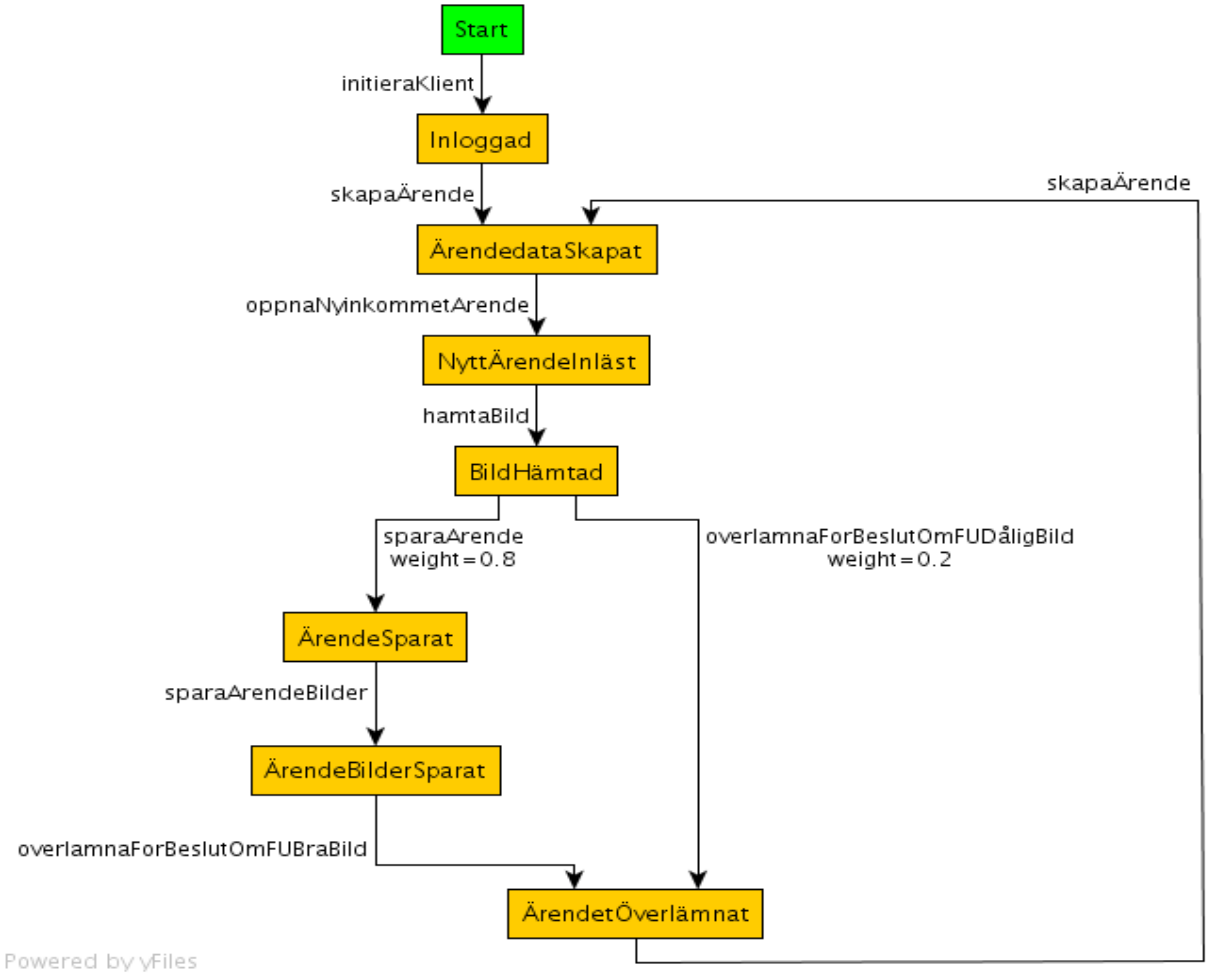
Systemet under test



Målet

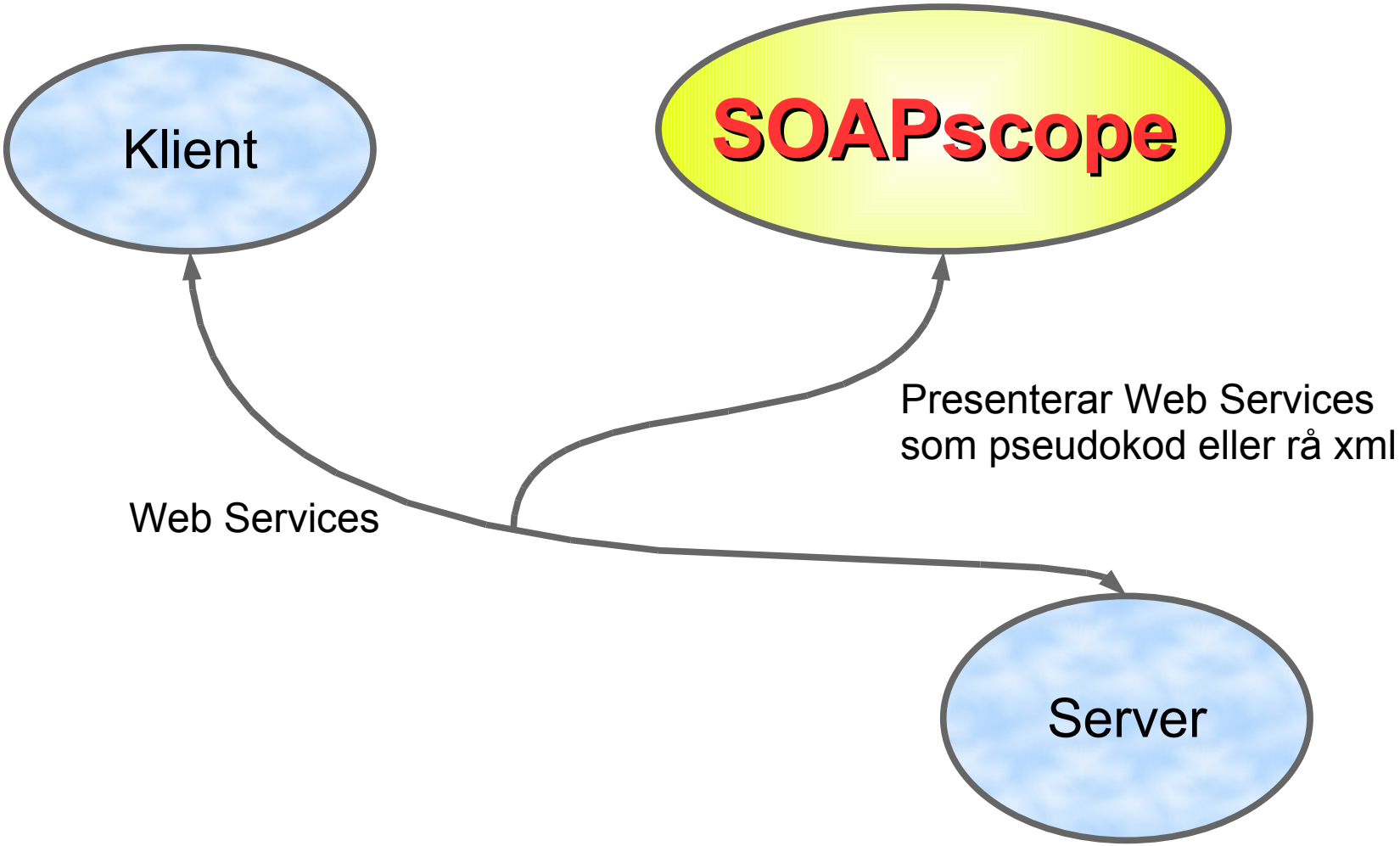
- Verifiera att arkitekturen höll prestandakraven.
- Snabbt testa många olika systemkonfigurationer.

Modellen



prolore

SOAPscope (Mindreef)



prolore

HttpClient (Apache Jakarta)

- Implementerar HTTP 1.0 samt 1.1 i Java
- Full implementering av samtliga HTTP metoder (GET, POST etc)

Implementationen - LoadRunner

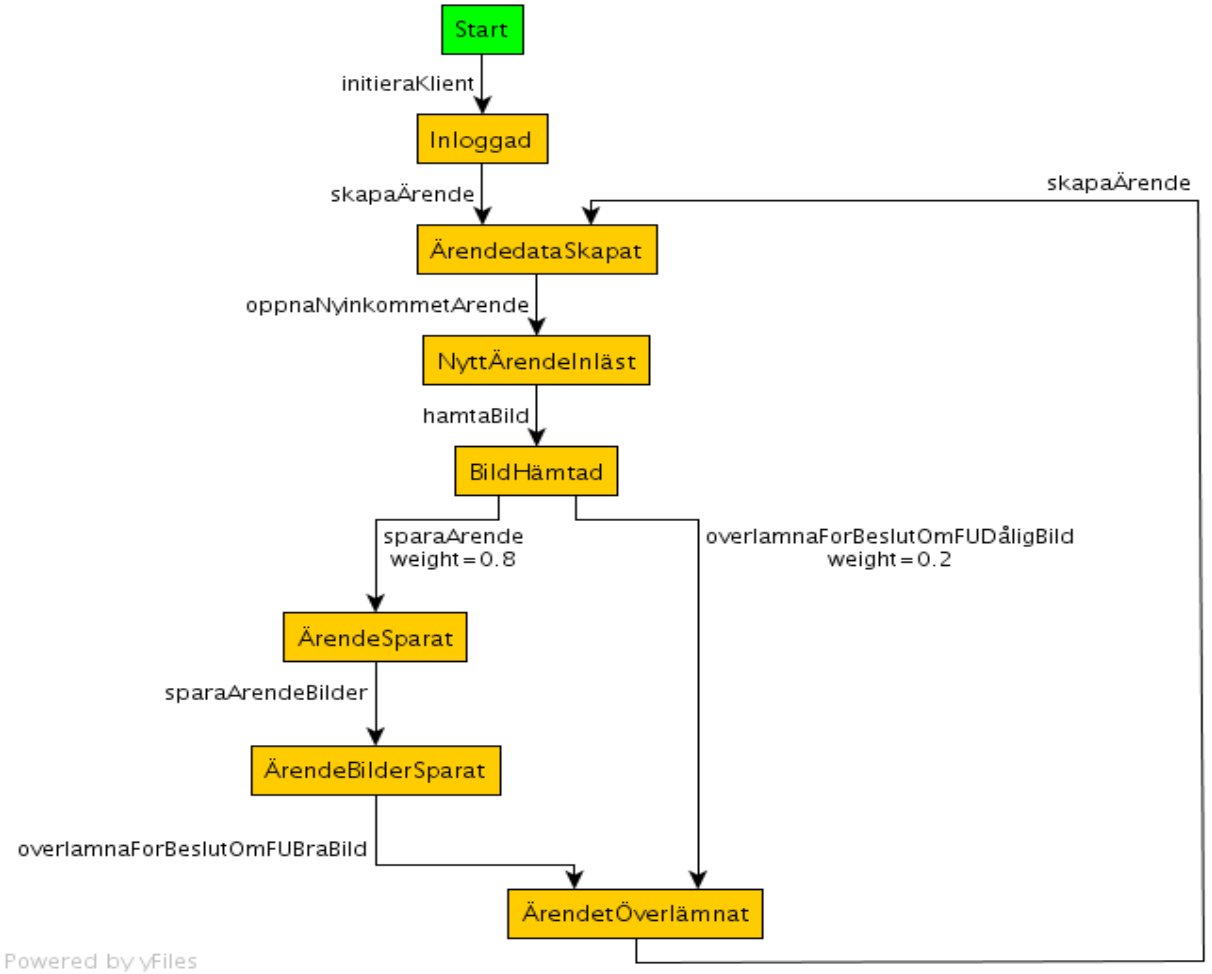
I Java Vuser skript, kan man placera vilken standard Java kod som helst.

(Gäller även för Corba-Java samt RMI-Java Vuser skript)

```
import lrapi.lr;

public class Actions {
  public int init() {
    return 0;
  }

  public int action() {
    return 0;
  }

  public int end() {
    return 0;
  }
}
```

prolore

Modellen



prolore

Implementationen - LoadRunner

Instansiering av ModelBasedTesting

Metoder genererade från tillståndsgrafen

```
import lrapi.lr;

public class Actions {

 public int action() {
  ModelBasedTesting mbt = new ModelBasedTesting(graphml,
                                                this);
  mbt.runrandomWalk( 3600 );
  mbt.runUntilAllVerticesAndEdgesVisited();
  return 0;
 }

 public void InitieraKlient() {
 }

 public void Inloggad() {
 }
}
```

prolore

Implementationen - LoadRunner

```
public void initieraKlient()
{
  String strXMLFilename = "soaprequest.xml";
  File input = new File(strXMLFilename);
  PostMethod post = new PostMethod(strURL);
  try {
    post.setRequestBody(new FileInputStream(input));
  } catch (FileNotFoundException e) {
    throw new RuntimeException("Could not load file: " + strXMLFilename);
  }
  post.setRequestHeader("Content-type", "text/xml; charset=utf-8");
  HttpClient httpclient = new HttpClient();

  result = 0;
  try {
    lr.start_transaction("initieraKlient");
    result = httpclient.executeMethod(post);
    lr.end_transaction("initieraKlient", lr.PASS);
    response = post.getResponseBodyAsString();
  } catch (HttpException e) {
    lr.end_transaction("initieraKlient", lr.FAIL);
    throw new RuntimeException("Transaction initieraKlient failed");
  } catch (IOException e) {
    lr.end_transaction("initieraKlient", lr.FAIL);
    throw new RuntimeException("Transaction initieraKlient failed");
  }
  post.releaseConnection();
}
```

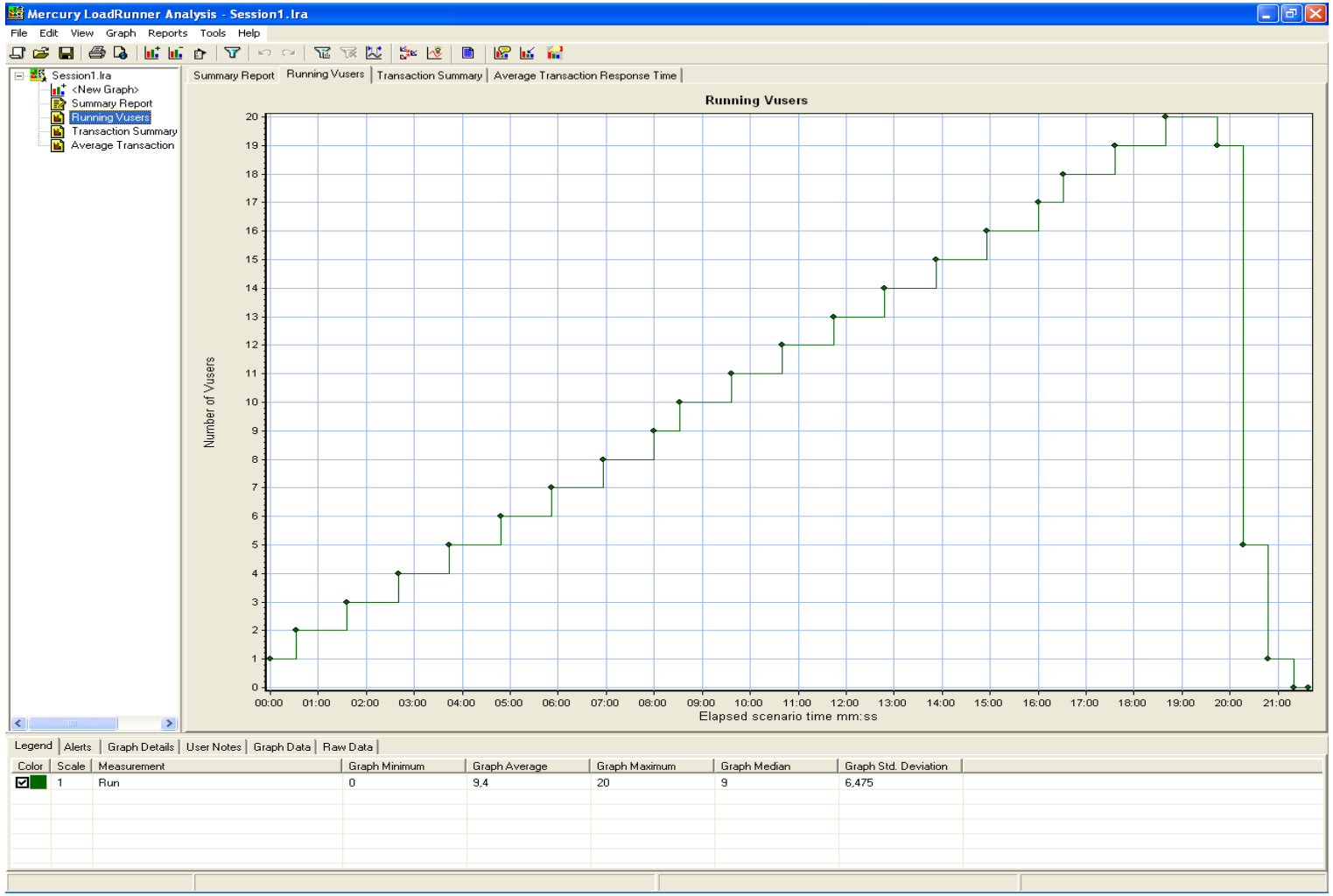
prolore

Implementationen - LoadRunner

```
public void Inloggad()
{
  Pattern p = Pattern.compile(
                  "<ns:initieraKlientResponse .*>.*</ns:initieraKlientResponse>",
                  Pattern.MULTILINE );
  Matcher m = p.matcher( response );

  logger.debug("Response status code: " + result);
  logger.debug("Response body: " + response );

  if ( result != 200 )
  {
    logger.error( "result: ” + result );
    throw new RuntimeExcection(”Verification failed”);
  }

  if ( !m.find())
  {
    logger.error( "response: ” + response );
    throw new RuntimeExcection(”Verification failed”);
  }
}
```

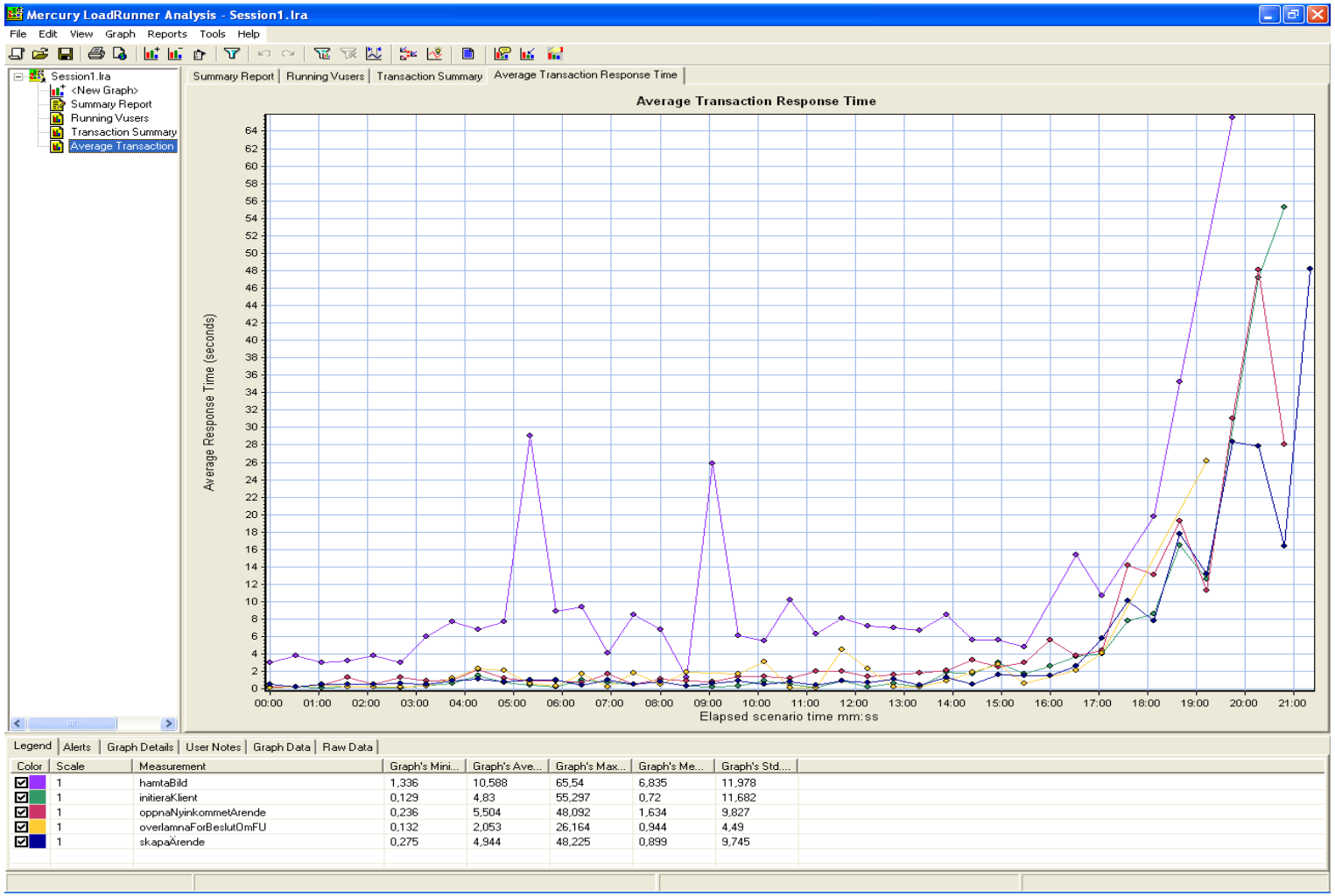
prolore

LoadRunner - Vusers



prolore

LoadRunner svarstider



prolore

Resultat – tail -f

```
tail -f mbt.log | grep ”Vertex: |\Edge: ”
```

```
2006-11-13 19:05:16,929 [main] INFO  Regression - Vertex: NyttÄrendeInläst
2006-11-13 19:05:16,939 [main] INFO  Regression - Edge: hamtaBild
2006-11-13 19:05:21,716 [main] INFO  Regression - Vertex: BildHämtad
2006-11-13 19:05:21,726 [main] INFO  Regression - Edge: sparaArende
2006-11-13 19:05:21,726 [main] INFO  Regression - Vertex: sparaArendeBilder
2006-11-13 19:05:28,055 [main] INFO  Regression - Vertex: ÄrendeBilderSparat
2006-11-13 19:05:28,446 [main] INFO  Regression - Edge: overlamnaForBeslutOmFUBraBild
```

prolore

Resultat – Statistik mbt

```
Test coverage edges: 33/263 => 12.547529%
Test coverage vertices: 26/189 => 13.756614%
Number of visited edges: 38
Number of visited vertices: 39
Execution time: 65 seconds
```

Resultat

- Snabbare ledtider, då vi ibland bara behövde ändra i modellen för ett nytt test
- Testade 50-tal konfigurationer under 2 veckor
- Designade om systemet. (WLI utelämnades, och vi sparade massa pengar!)

prolore

Summering

För- och nackdelar och framgångsfaktorer

prolore

Fördelarna med LoadRunner

- Enkelt att starta och stoppa tester med LR
- Använde oss av LRs monitorer för snabb återkoppling under testet
- Vår egen loggning med log4j för djupare analys av vårt eget system

prolore

Fördelar MBT i prestandatest

- Designa test direkt från användningsfallen
- Graferna gör det enklare att förstå testerna
- Förenklat underhåll
- Enkelt att omkonfigurera test

Nackdelar MBT i prestandatest

- Kräver javakunskaper
- Ovant arbetssätt/filosofi
- Många verktyg (ingen integrerad miljö)
  yEd => org.tigris.mbt => LoadRunner

Nycklar till succe

- Utbilda organisationen/projektet i MBT
- Demonstrera metodiken för utvecklarna
- Genomför ett pilotprojekt
- Utbilda testarna i java

prolore

Frågor

Presentationen finns att ladda ner på www.prolore.se

prolore

Resurser

Verktygen

http://mbt.tigris.org

http://graphml.graphdrawing.org

http://www.yworks.com

http://www.mercury.com/uk/products/performance-center/loadrunner/

http://www.mindreef.com

http://www.cygwin.com

Model-Based Testing

http://www.geocities.com/model_based_testing/

http://www.goldpractices.com/practices/mbt/index.php

Prestandatest

http://www.perftestplus.com/presentations.htm

prolore