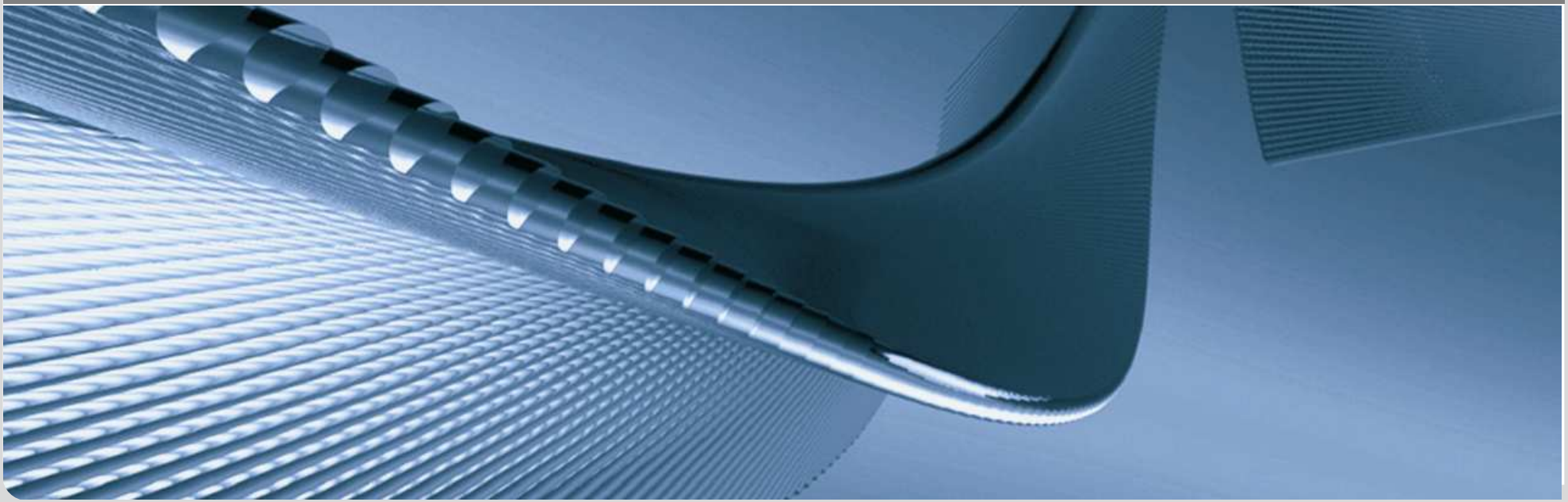
Karlsruhe Institute of Technology

The KeY-verified Verified Keyserver

Stijn de Gouw (Open University, NL), Mattias Ulbrich, Alexander Weigl | 27 April 2020

INSTITUTE FOR THEORETICAL INFORMATICS



Our program verifier KeY

Deductive verification

100% Java Card

Java Modeling Language (JML)

User Interaction Concepts

Modular Reasoning

Test case generation

Secure Information Flow

collaboration with TU Darmstadt and Chalmers University, Gothenburg

Modelling HAGRID in KeY

We present two formalisations of the HAGRID framework as spec'ed and verif'ed Java implementations:

The **automatic** model

- uses arrays to implement database and open requests
- specification on these arrays
- 70 loc, 90 los, 10 POs, **fully automatic**

loc/los = lines of code/spec, POs = # of proof obligations

The **map** model

- *uses map data structures to implement database and open requests*
- *specification on ADT maps*
- *"object singularities"*
- *146 loc, 262 los, 40 POs, **89 interactions***

Modelling HAGRID in KeY

We present two formalisations of the HAGRID framework as spec'ed and verif'ed Java implementations:

The **automatic** model

- uses arrays to implement database and open requests
- specification on these arrays
- 70 loc, 90 los, 10 POs, **fully automatic**

loc/los = lines of code/spec, POs = # of proof obligations

The **map** model

- *uses map data structures to implement database and open requests*
- *specification on ADT maps*
- *"object singularities"*
- *146 loc, 262 los, 40 POs, **89 interactions***

Modelling HAGRID in KeY

We present two formalisations of the HAGRID framework as spec'ed and verif'ed Java implementations:

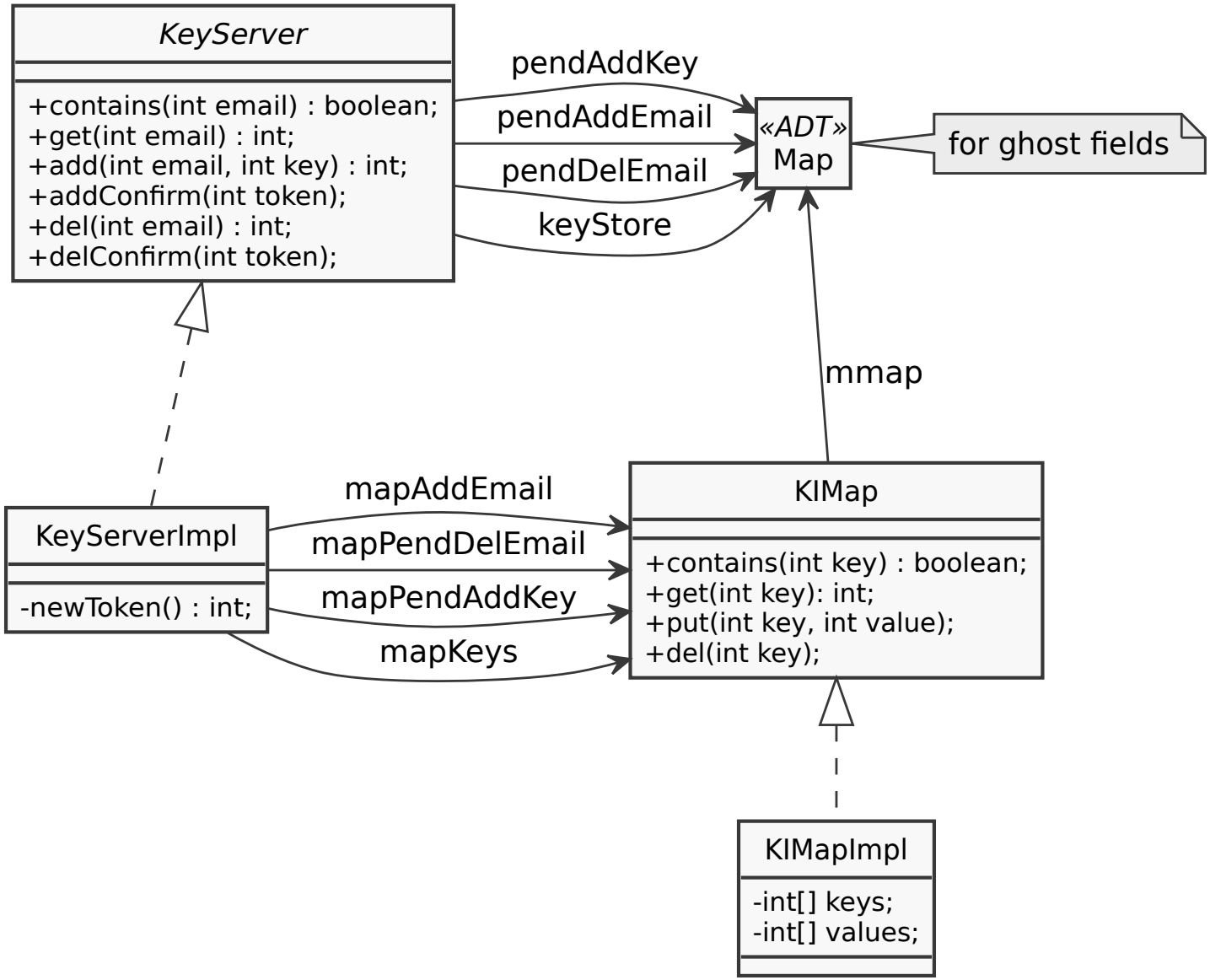
The **automatic** model

- uses arrays to implement database and open requests
- specification on these arrays
- 70 loc, 90 los, 10 POs, **fully automatic**

The **map** model

- uses map data structures to implement database and open requests
- specification on ADT maps
- "object singularities"
- 146 loc, 262 los, 40 POs, **89 interactions**

The map model



The map model

```
/*@ public normal_behaviour
  @  requires true;
  @  ensures keyStore == \old(keyStore);
  @  ensures pendAddEmail == \dl_mapUpdate(\old(pendAddEmail), \result, id);
  @  ensures pendAddKey == \dl_mapUpdate(\old(pendAddKey), \result, pkey);
  @  ensures pendDelEmail == \old(pendDelEmail);
  @  ensures !\dl_inDomain(\old(pendAddEmail), \result);
  @  assignable footprint;
  @*/
public int add(int id, int pkey);
```

The map model

```
/*@ public normal_behaviour
  @  requires true;
  @  ensures keyStore == \old(keyStore);
  @  ensures pendAddEmail == \old(pendAddEmail)[\result := id];
  @  ensures pendAddKey == \old(pendAddKey)[\result := pkey];
  @  ensures pendDelEmail == \old(pendDelEmail);
  @  ensures !\result \in \old(pendAddEmail);
  @  assignable footprint;
  @*/
public int add(int id, int pkey);
```