Towards DevOps for Privacy-by-Design in Data-Intensive Applications: A Research Roadmap
Context

- Growing interest in Big Data and data-intensive computing.
- **Data privacy** is a huge concern!
- Data privacy as a **primary quality aspect** for data-intensive applications.
Privacy Assurance

- **Privacy policy**: determining how third parties can access and use data.
- Defining, **monitoring** and enforcing privacy policies.
- Many existing privacy enhancing technologies:
  - Data anonymization
  - Encryption
  - **Attribute-based access control**
Our Solution

- **Tool prototype**: model-driven DevOps trace checking of temporal-based data access policies for data-intensive applications.
An Example Scenario

Update(SparkCluster, LinksPerPage, CassandraCluster) → P_{[20,2]}START
Modeling Language (1)

Inspired by Role Based Access Control and **SecureUML**.
Modeling Language (2)
Trace-Checking Service

Driver for Apache Cassandra

• Able to check permissions of type "Only within T1 and T2 any application running on cluster X can execute action A on column C of dataset D, where D is stored in Cassandra."
• Exploit Cassandra built-in tracing features.
• Build traces of events by querying Cassandra for the received queries.
Prototype Architecture
Threat to validity

• Trace checking Drivers have to provide a lot of system-level events.
• Checking privacy violations is not enough! Privacy for big data must be guaranteed.
• Are there other approaches rather than periodic trace checking that better fit the problem?
Future Work

• Systematic **evaluation** of the proposed solution.

• Towards privacy-aware **UML modeling**: exploit and extend SecureUML to express privacy policies.

• How to **react** to privacy violations?

• **Extended support** for privacy-awareness: continuous modeling and deploying of access control and data anonymization policies.
Towards a Research Roadmap

• The problem with privacy in big data is much bigger than this... We need to
  1. Understand and model new privacy aspects.
  2. Provide technologies and tools for privacy-by-design (e.g. formal verification of privacy models).
  3. Adapt traditional privacy enhancing technologies to the current technological environment.
  4. Extend big data technologies with built-in mechanisms to support privacy.
Q&A
MTL Metamodel
Recent Developments

- Complicated driver-based implementation.
- Simplify the trace-checking service and gain in flexibility at the expense of applications traspereancy.
- **Application instrumentation** with tracing features. No need for specific drivers to retrieve traces.
- Applications properly log events about their data accesses over which MTL formulae can be directly verified.
Conclusion

• A DevOps tool prototype to support the model-driven trace checking of privacy-aware data-intensive applications.

• Preliminary experimentation in industry looks promising.

• Definition of a research roadmap towards guaranteeing privacy in big data.