A Tracable Revocable Private Data Platform

Keywords
Data, traceability, revocation, ownership, RGPD

Supervisor: Stéphane Ducasse and Marcus Denker

Emails: stephane.ducasse@inria.fr and marcus.denker@inria.fr,

Context

Our society and industry manipulate enormous amounts of private data (with various privacy levels and properties). Medical personal data is the archetype of such data. Every day, individuals and doctors exchange and update data, but these sensitive updates are often traced nor controlled in ad-hoc fashion. In addition, the pharma industry is aggregating data from multiple sources to perform analyses on drugs and such data cohorts are consumed by different players. Within such scenario, corrupted data is expensive because it can render completely large campaigns invalid. However, once mixed is is impossible to trace and revoked corrupted data.

Such scenarios exhibit key properties that modern society is expecting from data manipulation:

- **Traced.** Any data should be able to request its origin, history and ownership. Even when recomposed to form new data, elementary data origin should be identifiable. In addition, the history of data changes should be accessible.
- **Revocable.** Invalid or corrupted data should be revocable and identifiable and this over the complete chain of composition.
- **Multi-facet data.** A single piece of data can have multiple facets (kind, dimension, validation, trigger...). In addition, such facetted data should also handle the fact that a piece of data can be nested and/or resulting from the composition of multiple other piece of data. Such resulting complex data should support revocation and origin/ownership of its subparts. Such facetted data may depend on the user and the user access rights: the same data in addition to its other properties may be invisible, read only, or writable.

We call **Smart Data** data having such properties (by opposition to big data). Such properties raise several challenges from an implementation point of view in terms of scalability (tracking changes or ownership), implementation (for example of warranty of immutability), and speed.

State of the art within the laboratory

Several international efforts have been carried around elements of the proposal but none of them on its entirety and in the context of the RGPD.

- **Capabilities.** Works such as the ones around E (the language), [**?**, **?**, **?**, **?**], introduced capabilities to restrain object properties to support isolation.
- **Isolation.** Caja is an attempt to isolate Javascript applications [**?**, **?**].
- **Ownership.** To support ownership, researchers investigated ownership types [**?**, **?**, **?**, **?**] for object encapsulation. [**?**] proposed different message sends to support dynamic ownership in a dynamically-type languages.

Work in the team. Nevertheless none of them radically rethink object-oriented programming in presence of the properties supporting traceability, ownership, history and revocation. We performed some preliminary study of the topic and developed a first prototype in the context of the CPER Data - 2 [**?**]. The current proposal will revisit such background and take advantage of the knowledge accumulated by the following PhD defended in the team:

- The PhD of C. Teruel entitled ”Adaptability and Encapsulation in Dynamically-Typed Languages: Taming Reflection and Extension Methods” and financed by the DGA was about the development of language constructs around ownership and how to control reflective operations that breach security concerns [**?**, **?**].
- The PhD of J.B. Arnaud entitled ”Towards First Class References as a Security Infrastructure in Dynamically-Typed Languages” was about the development of security constructs in dynamic languages [**?**, **?**, **?**].
- The work around proxies as part of the PhD of M. Martinez-Peck and C. Teruel [**?**, **?**]

Ph.D. Proposal

RGPD is a law to help citizens to control their data. However, there is a challenge to help developers build systems exhibiting by construction such properties. The easier it will be to produce systems like that the more chance we will have that our data are safely managed. **The goal of this PhD is to explore and design a language and execution engine where data exhibit properties that support RGPD and Smart Data.**

The PhD will develop language constructs and execution engine to support Smart Data.

The objectives are:

- Revisit and design object model and execution supporting the definition of multi-facet properties (frame KR-based).
- Define a model of elementary operations to compose, aggregate object while supporting warranty of no modification, history and origin.
- Revisit capability model to support revocation and the smart data properties.
- Define publication mechanisms with non modification warranty based on a block-chain back-end.
- Validate the results on real case studies.

PhD will work on:

- State of the art on revocation, capability, ownership, and support for RGPD.
- Identify scenarios with sensible data and complex multi player multi access rights.
- Evaluate first class instance variables and how they can contribute to the design of facetted data.
- Explore design of object-oriented languages supporting smart data (origin, history,...).
- Identify a minimal change algebra supporting smart data properties.
- Explore design on the light of scalability issues.

References

[BSBR03] Chandrasekhar Boyapati, Alexandru Salcianu, William Beebee, Jr., and Martin Rinard. Ownership types for safe region-based memory management in real-time java. In *PLDI '03: Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation*, pages 324–337, New York, NY, USA, 2003. ACM.

[CDNS07] Nicholas R. Cameron, Sophia Drossopoulou, James Noble, and Matthew J. Smith. Multiple ownership. In *Proceedings of the 22nd annual ACM SIGPLAN conference on Object oriented programming systems and applications (OOPSLA'07)*, pages 441–460, New York, NY, USA, 2007. ACM.

[CNP01] David G. Clarke, James Noble, and John M. Potter. Simple ownership types for object containment. In *Proceedings of the 15th European Conference on Object-Oriented Programming (ECOOP'91)*, LNCS, pages 53–76, London, UK, June 2001. Springer Verlag.

[CÖSW13] Dave Clarke, Johan Östlund, Ilya Sergey, and Tobias Wrigstad. Ownership types: A survey. In *Aliasing in Object-Oriented Programming. Types, Analysis and Verification*, pages 15–58. Springer, 2013.

[GN07] Donald Gordon and James Noble. Dynamic ownership in a dynamic language. In Pascal Costanza and Robert Hirschfeld, editors, *DLS '07: Proceedings of the 2007 symposium on Dynamic languages*, pages 41–52, New York, NY, USA, 2007. ACM.

[Mil06] Mark Samuel Miller. *Robust Composition: Towards a Unified Approach to Access Control and Concurrency Control.* PhD thesis, Johns Hopkins University, Baltimore, Maryland, USA, May 2006.

[MMF01] Mark Samuel Miller, Chip Morningstar, and Bill Frantz. Capability-based financial instruments. In *FC '00: Proceedings of the 4th International Conference on Financial Cryptography*, volume 1962, pages 349–378. Springer-Verlag, 2001.

[MS03] Mark S. Miller and Jonathan S. Shapiro. Paradigm regained: Abstraction mechanisms for access control. In *Proceedings of the Eigth Asian Computing Science Conference*, pages 224–242, 2003.

[MSL$^+$08] Mark S. Miller, Mike Samuel, Ben Laurie, Ihab Awad, and Mike Stay. Caja safe active content in sanitized javascript. Technical report, Google Inc., 2008.

[MYS03] Mark Miller, Ka-Ping Yee, and Jonathan Shapiro. Capability myths demolished. Technical report, Combex Inc, 2003.

[SD20] Ronie Salgado and Stéphane Ducasse. Towards a smart data processing and storage model. In *International Workshop on Smalltalk Technologies IWST'20*, August 2020.

[TEM$^+$11] Ankur Taly, Úlfar Erlingsson, John C. Mitchell, Mark S. Miller, and Jasvir Nagra. Automated analysis of security-critical javascript apis. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 363–378, Washington, DC, USA, 2011. IEEE Computer Society.