



Vaccinating Data against Abuse

PwC Cybersecurity Days 2020, Kurt Kammerer

DataVaccinator S.à.r.l.

- Vision “The Airbag for Data”
- Open Source Initiative (2019)
- Core Team
 - Volker Schmid, CTO (co-founder of regify)
 - Kurt Kammerer, CEO (co-founder of regify)
 - Open Source contributors
- Live with first customers
- Mission to mitigate Data Risks

Data Accidents are on the Rise

- 50,000 Ways to Loose Your Data
 - Theft, Hack, Leakage, Breach
- Value of Data increases
 - Enabler + Asset
- Regulation (GDPR..) and Compliance
 - Fines + Reputational Risks
- Rising Costs of Data Security and Privacy
 - Insurance + Protection + Data Loss Prevention



Solving the Problem at its very Root

Traditional Application

PID/PII + Contents



High Risk Data

DataVaccinator enabled Application

PID/PII

Personal identifiable data/info



Low Risk Data

Contents

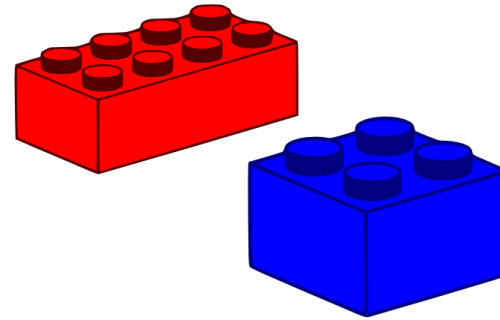


Low Risk Data

Privacy by Design

DataVaccinator USPs

- Pseudonymisation in realtime
- Software Lego Blocks
 - Developer friendly
 - Industrial quality
 - Minimal footprint
 - Any Application, even in IoT networks
- Commercially scalable
 - Industrial Approach
 - ROI for all stakeholders



Commercial Model

- DataVaccinator Revenues
 - License Support Services
 - Professional Services
 - SaaS for Data Services
- Scaling the Partner Community
 - IT Consultants and System Integrators
 - Compliance Experts and Law Firms
 - Software Vendors
 - Open Source Community



Vaccinate your Data against Abuse!

kurt@vaccinator.net

Appendices

DataVaccinator – for Local Applications

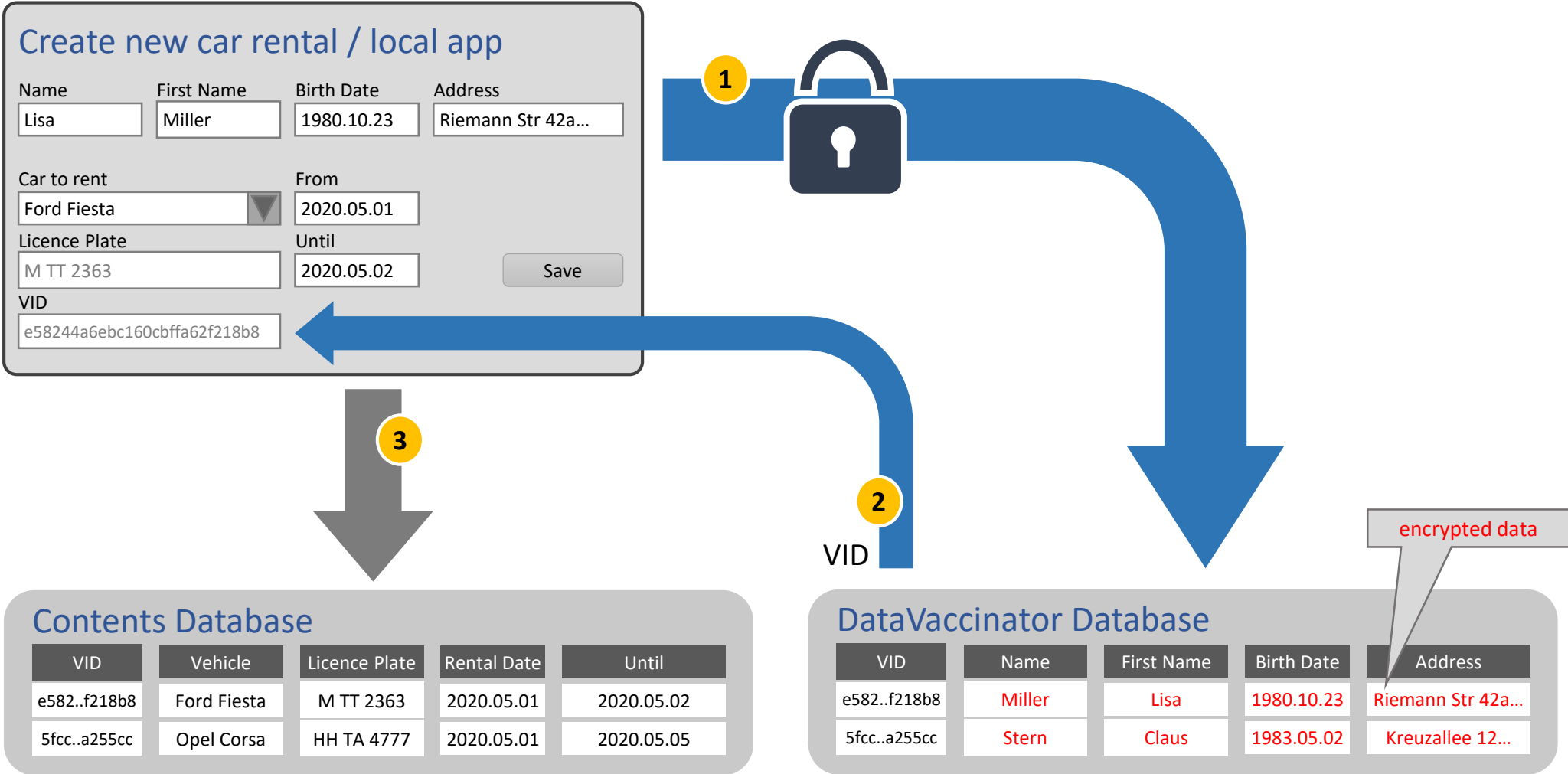


Figure: use case car rental (local app example)

DataVaccinator – for Web Applications

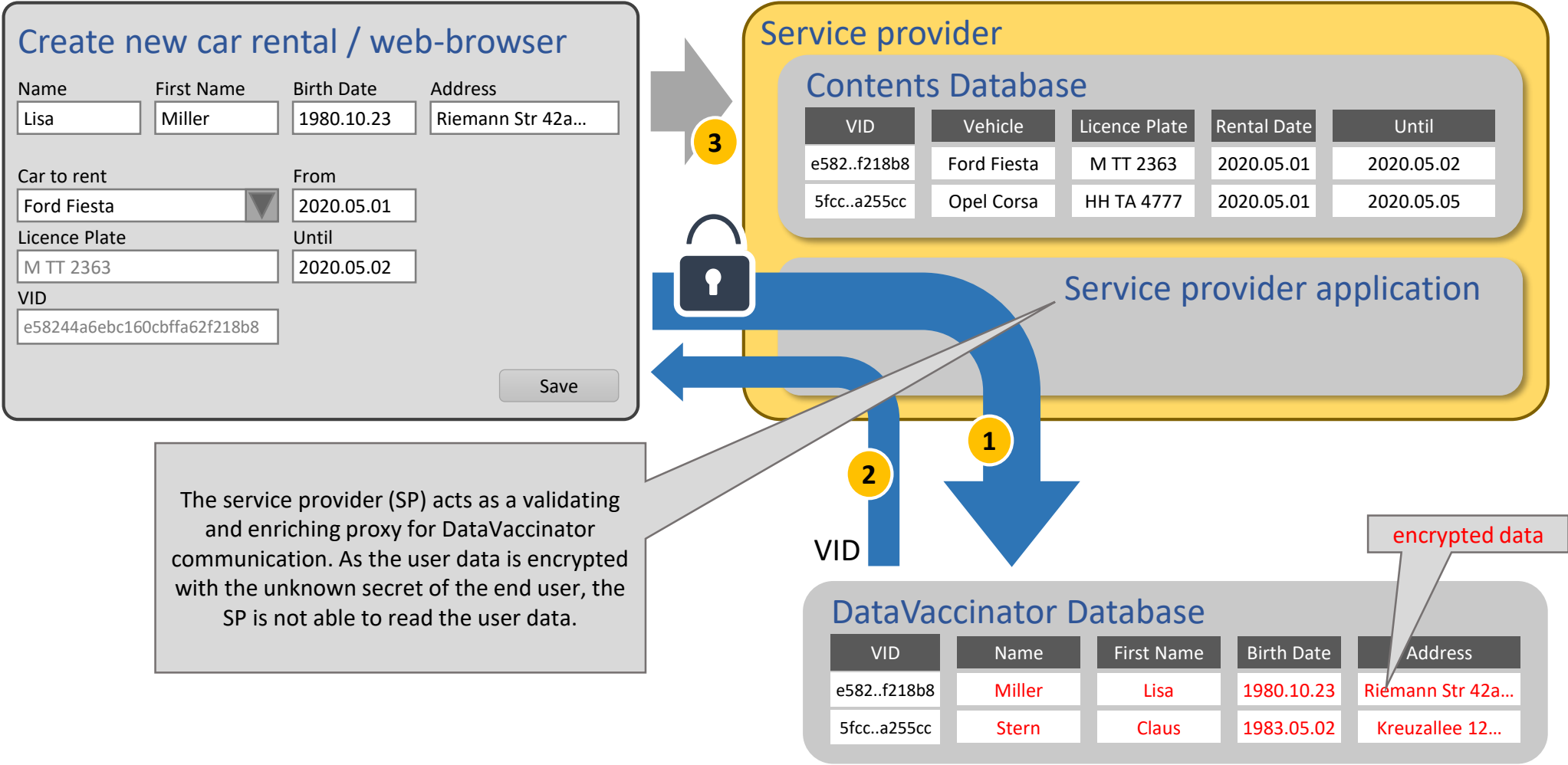
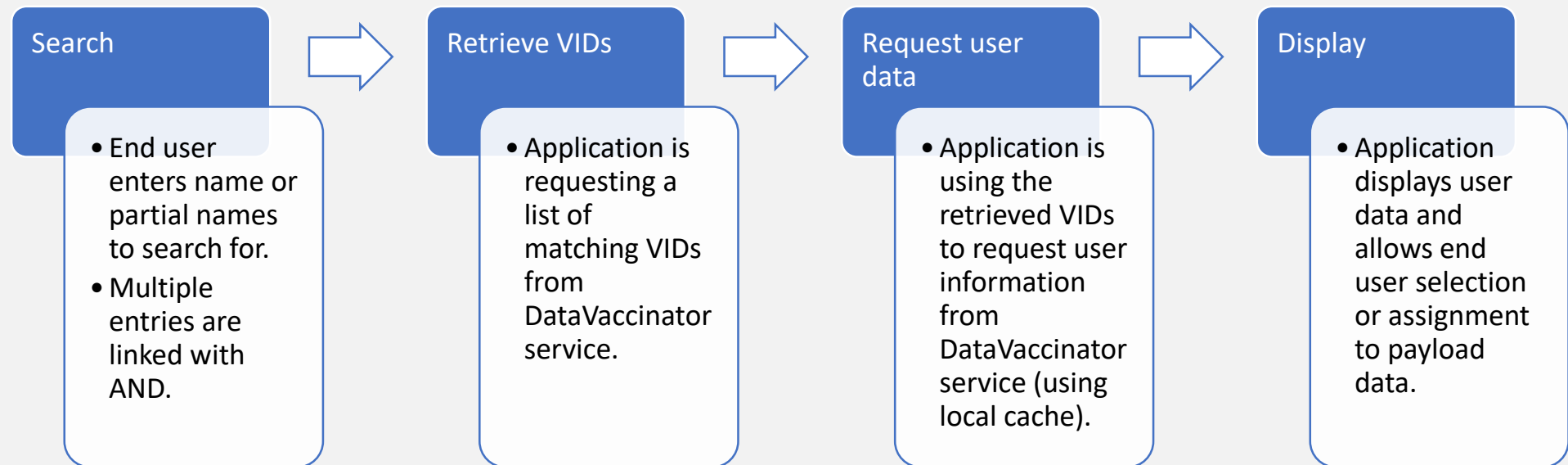


Figure: use case car rental (web app example)

DataVaccinator – Efficient Processing

Typical workflow for accessing data using DataVaccinator



Online searching within encrypted data for specific and partial tokens

Broad applicability as key tasks are enabled without compromising security.

DataVaccinator – Easy Integration

Example use of the JavaScript class: storing new user data

```
const appId = 'Rc-De_6nyCbb'; // Adapt to your needs

const providerUrl = 'https://vaccinator.de.regify.com/service.php';

// Create some example Vaccination Data
const vData = '{"firstname":"Spongebob","lastname":"Squarepants", '+
  '"Gender":"male","address_street":"Bikini Street", '+
  '"address_number":"42","address_city":"Bikini Bottom", '+
  '"address_zip":"12345", "address_country":"Pacific Sea", '+
  '}';

example(); // call example function

async function example() {
  try {
    // Create new instance of DataVaccinator class
    var v = new vaccinator();

    // Initialize with some service provider url, user, pwd and App-ID
    await v.init(providerUrl, 'volker', appId, 'password');

    // Enable search function using "firstname" and "lastname" vData fields
    v.enableSearchFunction( [ 'firstname', 'lastname' ] );

    // Upload and register Vaccination Data
    var vid = await v.new(vData);
    console.log('The new user Vaccination ID is ' + vid);

  } catch (e) {
    // catch any vaccinator class errors from here
    console.error(e);
  }
}
```

JavaScript example uploading a sample user dataset to the DataVaccinator service.

The JavaScript class takes care of encryption (AES256), generation of secure SearchHashes, server communication incl. authentication and local caching (using IndexedDB database).

The JavaScript class allows full asynchronous usage and is compatible with web browsers and also node.js.

Modules and libraries for other environments are in preparation.