A Framework for Automated Verification in Software Escrow

Stephan Strodl

IPRES 2013

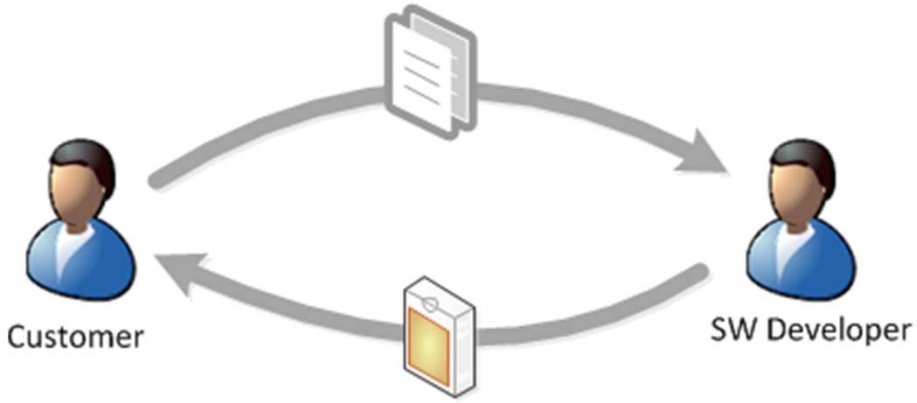
Lisbon, Portugal

Introduction

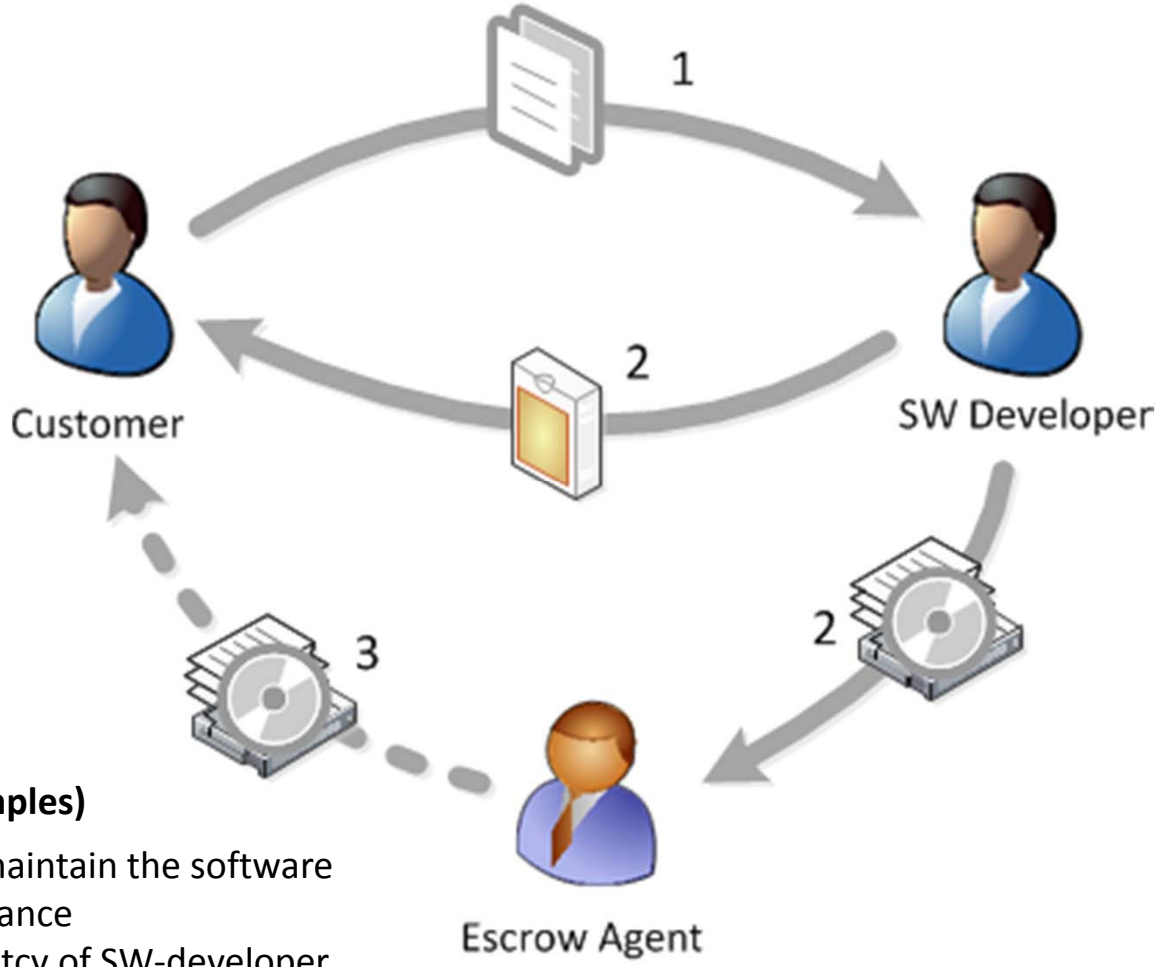
- Businesses need external knowledge and services (e.g. in form of software)
- Two possibilities:
  - Integrate knowledge provider (acquisition)
  - Outsourcing
- Software essential assets

Risks of Outsourcing

- Lost of control
- Financial standing of the vendor
- Sale of the vendor
- Maintenance and adjustments of the system
- Business critical application



Software Escrow



Release events (examples)

- Failure to support/maintain the software
- Insufficient maintenance
- Insolvency/ bankruptcy of SW-developer

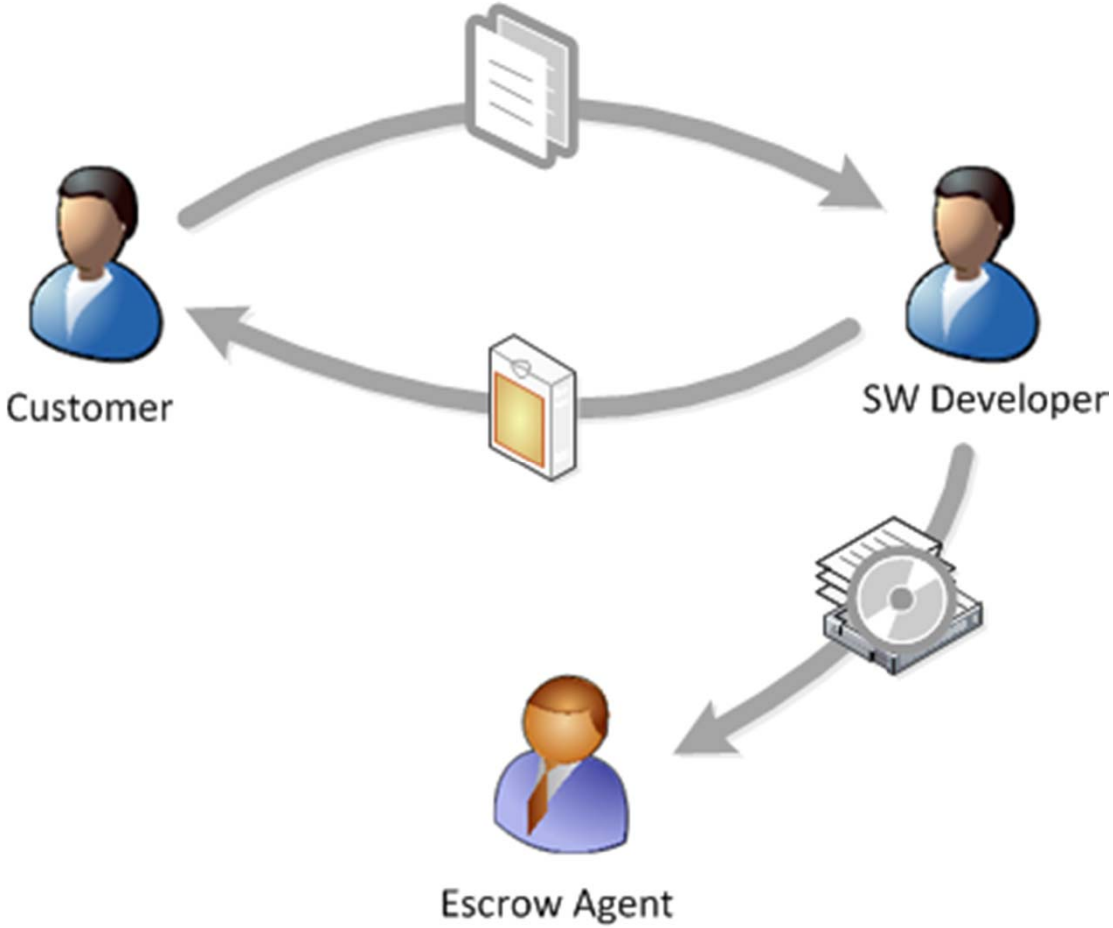
Escrow parties

- Customer
  - Use of the software
  - Protection of investment
- SW Developer
  - Provides object code to consumer
  - Commits source code to Escrow Agent
  - Obligations to the consumer as well as the Software Escrow Agent
- Escrow Agent
  - Verification of deposit material
  - Deposit of source code
  - In case of agreed circumstances release of the source code to the consumer

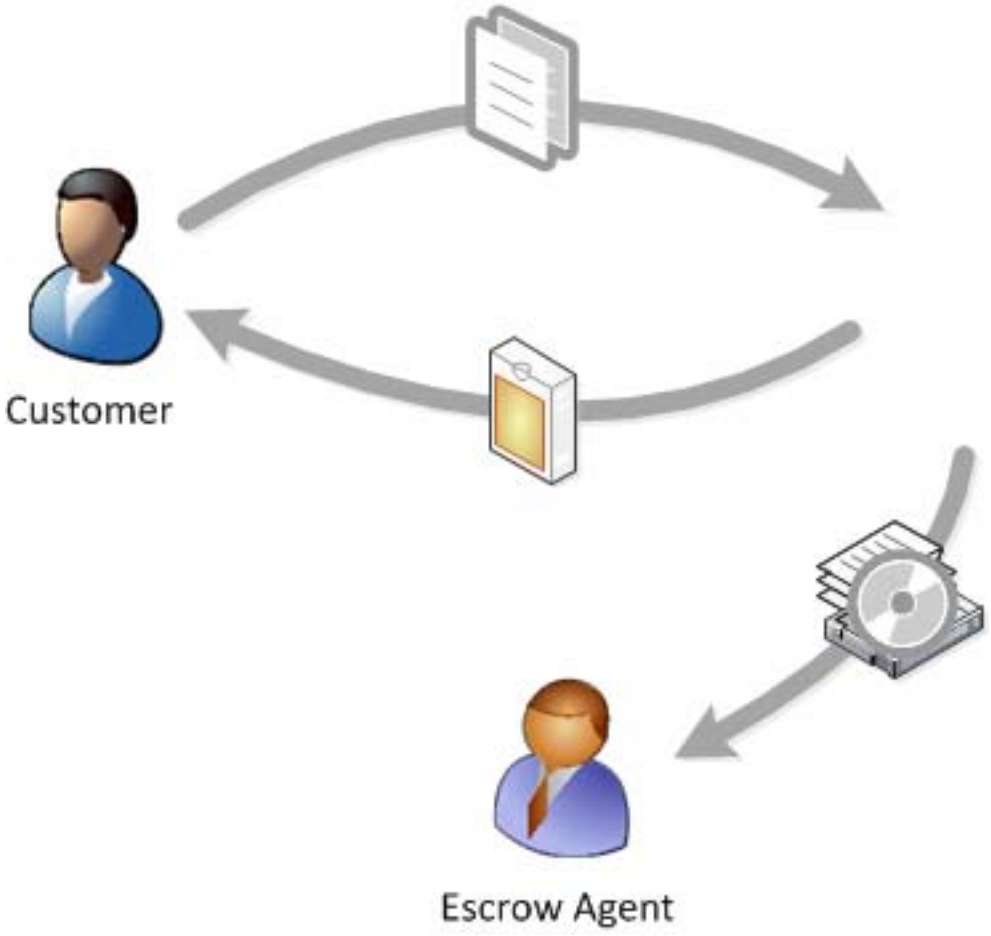
Motivation of parties

- Customer
  - Risk mitigation strategy
    - Maintenance and support
    - Protection of business
  - Hedging of investments
  - Verification of source code
- SW-Developer
  - Evidence of copyright
  - Warranty claim
  - Confidence-building measure
  - Payment

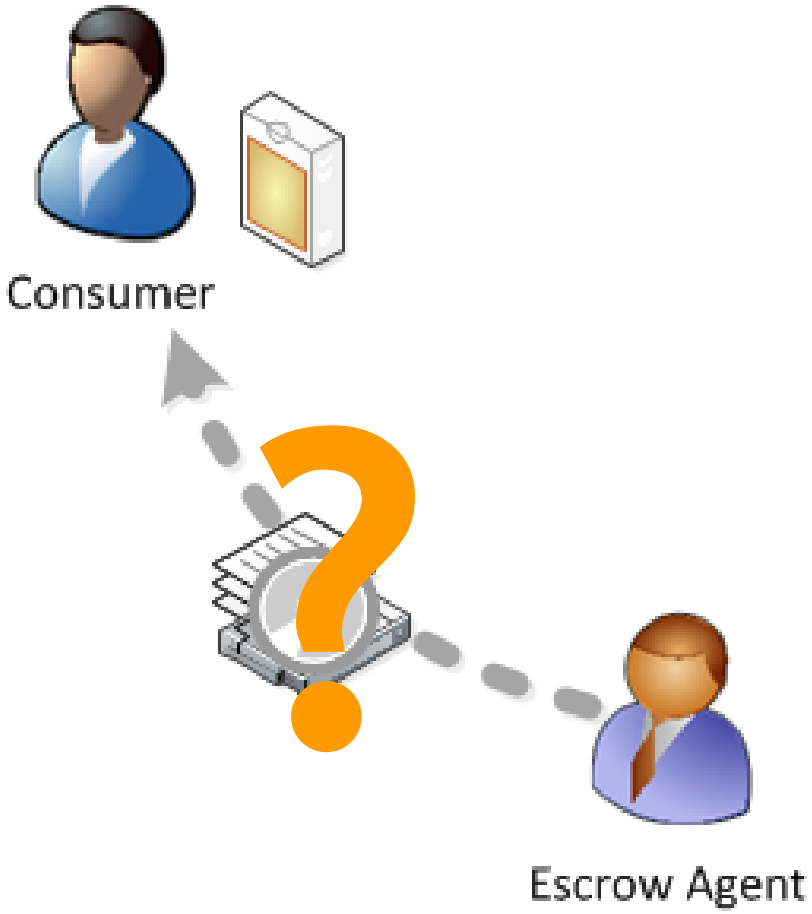
Escrow Case



Escrow Case



Escrow

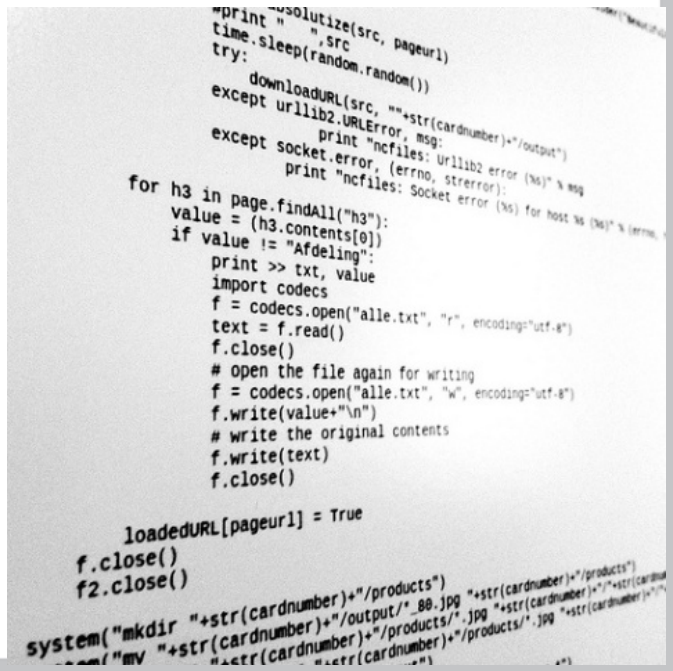


Escrow Risks

- Read error storage media
- Source code incomplete
- Build environment not deposited
- Configuration is not available
- Instructions missing
- Test data missing

Escrow Risks

- Documentation insufficient
- Licences and rights not included (Development- & Build- Environment, Libraries)
- Deposited material is not up to date
- Intentional source code hiding



Requirements

- Completeness
  - Source code only part of a software
  - Without additional information almost impossible to understand, analyse, use and change the source code
- Quality
  - Up to date
  - Maintainability

Three phases of Escrow

- Phase I – Planning
  - Agreement on contract specifying measurements for Software Escrow
- Phase II – Execution
  - Deposit of material including iterations
- Phase III – Redeployment
  - Release of material if trigger event occurs

Phase I - Planning

- Selection of the escrow agent
  - Confidence of consumer and SW developer
  - Requirements for IT infrastructure
  - Data security requirements
  - Information-/notification obligation
- Alignment of the contracts
  - Inclusion of basic escrow provisions in the software licensing/maintenance contracts
- Select deposit material

Phase I - Planning

- Rights to the software artefacts
  - Rights of the escrow agent
  - Rights of the software developer
- Clarification of the legal consequences of the release
  - Determination of the rights transferred to the consumer
  - Obligations on confidentiality
- Setting up an Escrow contract

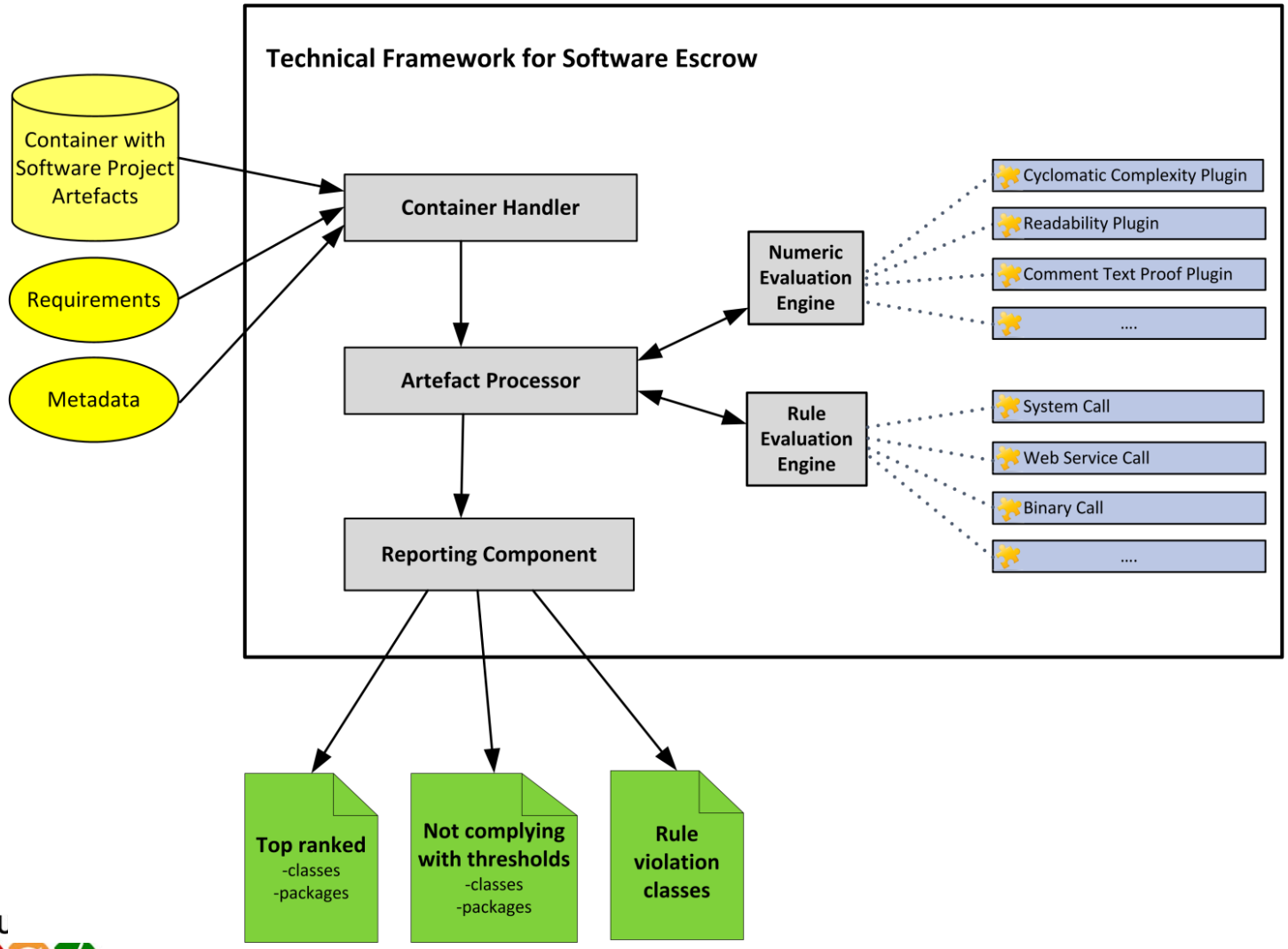
Phase II - Execution

- Deposit procedure
  - Submission deadlines
  - Existing verification procedures
  - Thresholds for successful deposit
  - Information-/notification obligations
- Verification of material
  - Technical Software Escrow Framework

Phase III - Redeployment

- Release of materials
  - Release events (insolvency, refusal to maintenance, …)
  - Release process
- Information-/notification obligations

Technical Framework - Overview



SW Quality


ISO 25010
portability
adaptability
installability
replaceability
maintainability
modularity
reusability
analysability
modifiability
testability
reliability
maturity
availability
fault tolerance
recoverability
usability
appropriateness recognisability
learnability
operability
user error protection
user interface aesthetics
accessibility
functional suitability
functional completeness
functional correctness
functional appropriateness
security
confidentiality
integrity
non-repudiation
accountability
authenticity
performance efficiency
time behaviour
resource utilization
capacity
compatibility
co-existence
interoperability

Technical Framework - Plugin groups

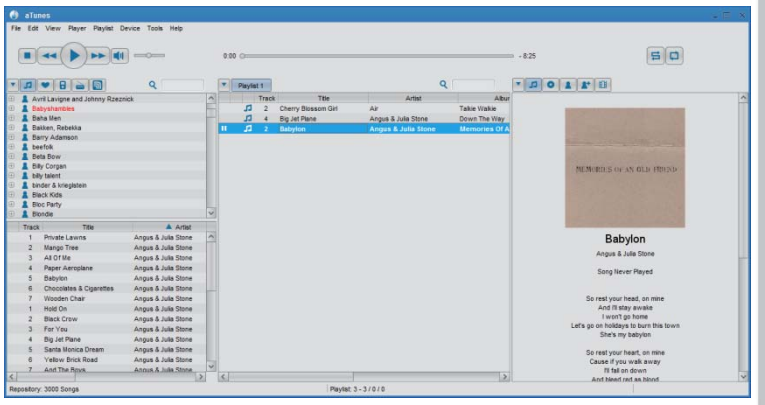
- Build environment
- Quality metric measurements
- Test environment
- Source code comments
- Specification verification
- System documentation
- Data sources
- External references
- Code obfuscation

Checks for Validation

- Build & Release
- Comments
  - Language detection
  - Comment text proof check
  - Comment ratio
- Complexity
  - Complexity calculation (McCabe, Halstead)
  - Comment-complexity ratio
- Nonstandard libraries
  - Hiding of source code
- Binary and external calls
  - Dependencies and hiding of source code
- License check

Use Case

aTunes 3.0.8



- open source audio player
- JAVA
- Good documentation
- Web Service and binary calls

[Technical Framework]