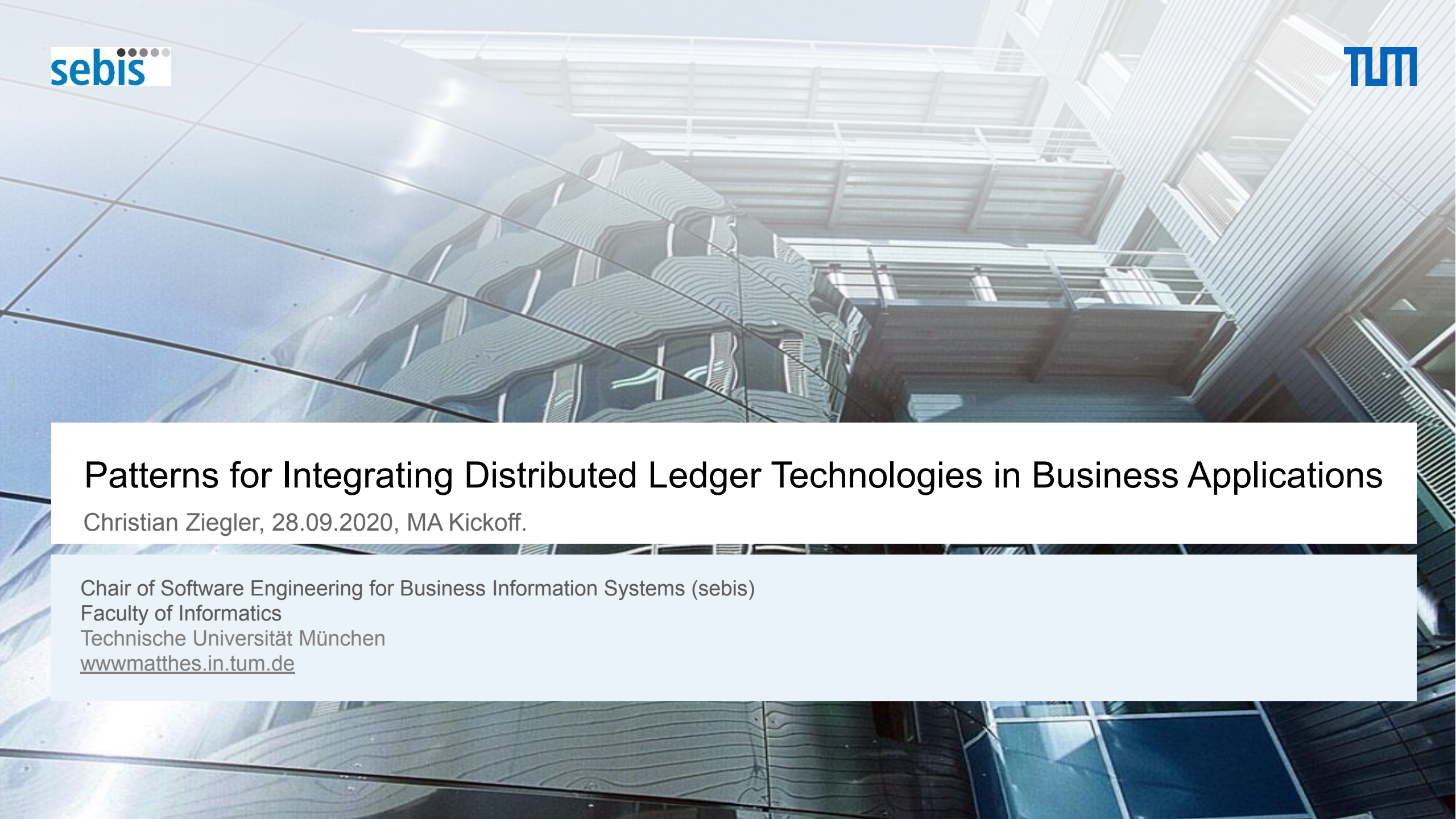


Patterns for Integrating Distributed Ledger Technologies in Business Applications

Christian Ziegler, 28.09.2020, MA Kickoff.

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Outline

Motivation
- A Brief History of DLT
- Why we need Patterns

Research Approach
- Overview
- Finding Projects
- Related Work
- Research Questions

Patterns
- Possible Pattern Categories
- Possible Pattern inside the Categories

Motivation - A Brief (and extremely simplified) History of DLT

Bitcoin

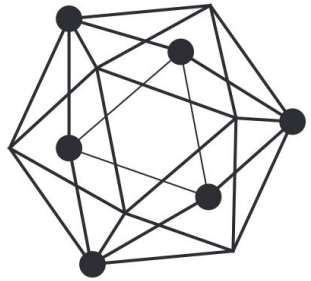
“Transfer money without a third party”

7187 Cryptocurrencies are active today [1]

Ethereum

“Execute code on a global virtual machine”

More than 2 million Smart Contracts deployed on Ethereum [2]



Hyperledger

“Build your own private blockchain for your enterprise system with all its benefits (and downsides...)”

Hyperledger collaborates with more than 250 companies [3]

[1] https://coinmarketcap.com/all/views/all/
[2] https://etherscan.io/accounts/c/2
[3] https://www.hyperledger.org/

Motivation - Why we need Patterns

Blockchain is new technology that struggles finding its way into the market

Companies have many pilot applications that contain blockchain-technology

Very few pattern catalogues have been published on DLT

developer teams tend to “re-invent the wheel” regarding to blockchain-technology due the immaturity of the technology

Flaticon.com: Pixel perfect, Freepik, mynamepong, Good Ware

Outline

Motivation
- A Brief History of DLT
- Why we need Patterns

Research Approach
- Overview
- Finding Projects
- Related Work
- Research Questions

Patterns
- Possible Pattern Categories
- Possible Pattern inside the Categories

Overview

- Look at the market and the trends and find projects
- Find related work such as existing published DLT pattern catalogues

Mostly finished

- Look at the technology that the projects use
- Find out which technologies are used most and extract implementations
- Convert the extracted implementations into the (adapted) GoF pattern description

TODO

Research Approach - Finding Projects

Google Scholar

Searches for “DLT”, “Distributed Ledger Technology”, “Blockchain” sorted by newest -> Mostly “hard to argue blockchain projects”

Scopus

Searches for published pattern catalogs: Resulted in one big pattern catalogue

Checked out top 10 projects: They deliver the base technology most DLT projects use

Everest.link

Newest hype projects and very small projects with extremely niche ideas

Research Approach - Related Work

(Just one Paper?!)

A Pattern Collection for Blockchain-based Applications [1]

- Interactions with external world patterns
  - **Oracle**: Introducing the state of external systems into the closed blockchain execution environment
  - **Reverse Oracle**: The off-chain components of an existing system rely on smart contracts running on a blockchain to supply requested data and check required conditions.
  - **Legal and smart contract pair**: A bidirectional binding is established between a legal agreement and a corresponding smart contract.
- Data Management
  - **Encrypting on-chain data**: Ensure confidentiality of the data stored on blockchain by encrypting it.
  - **Tokenization**: Using tokens on blockchain to represent transferable digital or physical assets or services
  - **Off-chain data storage**: Use hashing to ensure the integrity of arbitrarily large datasets which may not fit directly on the blockchain
  - **State channel**: Transactions that are too small in value relative to a blockchain transaction fee or that require much shorter latency than can be provided by a blockchain, are performed off-chain with periodic recording of net transaction settlements on- chain.
- Security
  - **Multiple authorization**: A set of blockchain addresses which can authorise a transaction is predefined. Only a subset of the addresses is required to authorize transactions.
  - **Off-chain secret enabled dynamic authorization**: Using a hash created off-chain to dynamically bind authority for a transaction.
  - **X-confirmation**: Waiting for enough number of blocks as confirmations to ensure that a transaction added into blockchain is immutable with high probability
- Structural Patterns of Contract
  - **Contract registry**: Before invoking a smart contract, the address of the latest version of the smart contract is located by looking up its name on a contract registry
  - **Embedded permission**: Smart contracts use embedded permission control to restrict access to the invocation of the functions defined in the smart contracts.
  - **Data contract**: Store data in a separate smart contract.
  - **Factory contract**: An on-chain template contract is used as a factory that generates con- tract instances from the template.
  - **Incentive execution**: A reward is provided to the caller of a contract function for invoking it

Most are interesting for my research but with different categories

[1] XU, Xiwei, et al. A pattern collection for blockchain-based applications. In: *Proceedings of the 23rd European Conference on Pattern Languages of Programs*. 2018. S. 1-20.

Research Approach - Research Questions

RQ1 What are common features that Distributed Ledger Applications share?

RQ2 How can these features be put in a formal pattern definition?

RQ3 From the defined patterns, which are the most important ones?

Outline

Motivation
- A Brief History of DLT
- Why we need Patterns

Research Approach
- Overview
- Finding Projects
- Related Work
- Research Questions

Patterns
- Possible Pattern Categories
- Possible Pattern inside the Categories

Possible Pattern Categories

Unique Asset Storing	Unique Asset Transfer	On-Chain Code Execution	Identity Management
Information Broadcasting	Off-Chain Data Storage	On-Chain Data Storage	Decentralized Exchange
Governance	Cryptocurrency Minting	Cryptocurrency Storing	Cryptocurrency Transfer

Possible Patterns inside the Categories

Identity Management	Decentralized Exchange	Cryptocurrency Minting	Cryptocurrency Storing
• Asymmetric security keys (Bitcoin, Ethereum, …) • Certificate based authentication (Hyperledger MSP, …)	• Atomic swaps (Bitcoin, Bitcoin Cash, Decred, Litecoin, …) • Flash Swaps & Liquidity (Uniswap)	• Pre-Mine • Mining (PoW, PoS, …) • Minting (Stablecoins) • Elastic Supply (AMPL)	• UTXO based (Bitcoin, Litecoin, …) • Account based (Ethereum, BnB, …) • Hybrid approach (Accounting systems on top of UTXO based coins. e.g. Bitcoin)

Christian Ziegler
christian.ziegler@tum.de

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for Business Information Systems

Boltzmannstraße 3
85748 Garching bei München

Tel +49.89.289. 17132
Fax +49.89.289.17136

matthes@in.tum.de
wwwmatthes.in.tum.de

Discussion

- Are there other important sources for projects and pattern catalogues that I have missed?
- Are the research questions conclusive and expedient?
- Do you think that my pattern categorisation makes sense?
- Do you think that the example patterns I listed for four categories would make good patterns?
- In general would you agree that the direction I am aiming regarding the categories and pattern is productive?