

WHITE PAPER

Version 1.0

rell.CHain

Swiss Permissioned Enterprise Blockchain

PERMISSIONED BLOCKCHAIN ON C#

The Enterprise Blockchain System

The rell.CHain was developed to secure data and systems while maintaining full control, both technically and legally, within its own organization or consortium. With this multi-node blockchain, nodes can be distributed and operated in the local corporate network and on private clouds. Control over these own nodes is maintained even if the hardware is distributed to separate data centers. Building and operating your own robust and secure IT infrastructure creates trust and is a critical competitive advantage in a digital economy.

😪 Services							
File Action View	Help						
🗢 🔿 🔚 🖾 🔒 🛛 🖬 🕨 🔳 II ID							
Services (Local)	Name	Description	Status				
	_rellCHain_miner_000001	Blockchain MinerService Rell.IO AG	Running				
	🧟 _rellCHain_miner_000002	Blockchain MinerService Rell.IO AG	Running				
🤹 _rellCHain_miner_000003		Blockchain MinerService Rell.IO AG	Running				
	ActiveX Installer (AxInstSV)	Provides User Account Control validati					
	AllJoyn Router Service	Routes AllJoyn messages for the local	3				

Special feature of rell.Chain

The implementation of rell.CHain is based on .**NET Core** and the programming language **C#**. Rell.CHain does not require complex libraries or frameworks from third parties. For this reason, the system requirements for hardware and operating system are very low and resource-saving. A very high performance with extremely short reaction times is achieved on current server systems with correspondingly powerful hardware. Since it is a private blockchain, the block time can be kept very low (block time ≤ 1 sec).

BUSINESS-CASE

The optimal SETUP

The behavior of rell.CHain can be optimized for your own specific business case by parameterization and the corresponding SETUP. This blockchain is extremely scalable and the transaction rate is primarily determined by the performance of the operating system and the hardware.

Performance

An average performance of 5,000 transactions per second is possible.



OPERATING MODES AND USE CASES

Scalable

The behavior of rell.CHain can be optimized to different requirements by parameters. For your own specific business case, the correct SETUP is determined and distributed to all nodes.

SETUP-Mode	Use Case	Business Case	Requirements
Tracking	Tracking Logging Archiving Measuring Metering	Auditing Record Management Accounting Quality Insurance IoT-Monitoring	Complexity: low Transaction-Count: > 10.000 Confirmation-Time: > 60 sec
Ledger	Confirmation Verification Acceptance Certification Unique-Identifier	Banking Insurance Logistics Trading-Platform Identity-Management IoT-Control	Complexity: high Transaction-Count: > 1.000 Confirmation-Time: > 15 sec
Switch	Lock/UnLock Switch On/Off Authentication Approvement Check	Banking Insurance Logistics IoT-Control	Complexity: low Transaction-Count: >= 1 Confirmation-Time: < 1 sec
DB-Reference	Delete Data on DB- Reference Interfaces: - Oracle - SQL-Server	Banking Insurance Traffic Trading-Platform	Complexity: high Transaction-Count: > 1.000 Confirmation-Time: > 15 sec

rell.io

TECHNICAL FACTS

SETUP-Parameter

Parameter :	Value	Unit	Default / Range
Blocksize :	value	KB	(8-32.000)
Merkle-Path:	value	Hashes	(4-16)
Transaction-Count:	lower limit		(1-1.000)
	target		(10.000)
	upper limit		(200.000)
Confirmation-Time:	lower limit	sec	(1)
	target	sec	(20)
	upper limit	sec	(300)
Difficulty:	value	hex	(0x00FFFFF)

Type:private permissioned multi-node-blockchainConsensus:proof of work - based on SHA-256 (difficulty determined by SETUP)Storage:digitally signed immutable data blocks structured in merkle treesTransaction Fee:0.00 USD/EUR (gas price determined by SETUP)Transaction Rate:on average 5.000 transactions per secondSETUP:(maxBlockTx=10.000, Confirmation-Time=1 sec

Hardware and Operating System Requirements

.NET Core is a general-purpose development platform maintained by Microsoft and the .NET community on <u>GitHub</u>. It's cross-platform (supporting Windows, macOS, and Linux) and can be used to build device, cloud, and IoT applications. The **rell.CHain** "**C#** .**NET Core implementation**" runs on high-performance servers, on standard hardware (e.g. desktop PCs) and even on microcontroller systems (embedded systems / embedded systems / single-chip computer systems).

SCALABLE APPROACH

block formation concept

Version 1.0 of rell.CHain is designed for 4 operating modes in which blocks are formed in different ways. The general differences result from the requirements:

- record large amounts of data
- secure business or production processes
- short reaction times for binary decisions
- Conformity with the basic EU data protection regulation (DSGVO/GDPR)

Record large amounts of data

In tracking mode, the system is optimized to store very large amounts of data in the blockchain in a revision-proof manner.

The recording of production measurement data or batch movements generates enormous amounts of data. Transactions must be confirmed securely, but not very fast.This special feature of the requirements in this area can be seen, for example, in "metering" when the energy consumption must be recorded in a tamper-proof manner. However, since billing takes place at different times, speed is not the issue here.

The conceptual approach for this operation mode consists of collecting transactions in a staging area and processing and compressing them in the form of candidate Merkle blocks. Transaction processing and block formation are quantity-oriented. This operation mode is suitable for supporting QA and document management systems.



rell.io

Secure business or production processes

Blockchain technology optimizes operations or production processes, changing the way we do business. The standard "Ledger" mode of rell.CHain enables audit-proof transactions across organizational and network boundaries. This enables the automation of business cases and production steps (loT, Industry 4.0) as well as accelerated processing of international payment transactions. The secure interaction with third-party providers and end customers can be implemented via platforms.



Short response times for binary decisions

Due to the slim structure, the essential header information can be kept in evidence on all nodes. In this mode, a new transaction is confirmed immediately after a few checks of essential key information. This allows a response time of less than 1 second for a binary decision (lock/unlock, on/off, approved/rejected). This mode enables secure switching operations on micro-controller systems (loT) or verified approval in a business release process (e.g. payment release).



Conformity with the EU data protection basic regulation

The basic data protection regulation (DSGVO, or GDPR: General Data Protection Regulation) came into force throughout the EU in 2018 and most companies must take additional security measures to meet all requirements.

With regard to the processing of personal data, the GDPR includes, among other things, the "right to be forgotten" and thus the deletion of data. Because all data is stored in the blockchain, nothing can be deleted. In the operating mode "DB-Reference" the rell.CHain works together with a relational database (RDBMS interface). By referencing the corresponding data records and a close coupling on system level, the regulations of the GDPR can be kept. The advantages of the two database technologies RDBMS and Blockchain are combined and thus used very effectively for the "benefit of the customer".

BLOCKCHAIN DATA HUP

SECURE STORAGE OF OWNERSHIP RECORDS: DIGITAL IDENTITY, ACCOUNTS, TRANSACTION

Node 1 • Chain, SETUP • Chain, SETU

BUSINESS APPLICATION

BLOCHCHAIN SYSTEM



[1] A new Input-Transaction was received by one API-Server – and will be synchronized to every other node at first of all.

[2] On every node the Input-Mining-Pool will be created by local MemPool-Service. The mining process will be started on each node at the same time.

[3] The inter process communication will be handled on a peer2peer network. After successful verification a "accept message" will propagate and Block #38 will be created.

TECHNICAL DETAILS



[1] Valid Token: mTab – Known peers only, are accepted in the communication process.

[2] Encrypt: keyTable – PublicKey-Encryption of outgoing messages. PrivateKey-Decryption makes a message useful for one target receiver only.

[3] Signed Data: The signature of a message will be verified by the receiver, to ensure the information was created by a known sender.

SEQUENCE DIAGRAM



[1] The Blocktime Configuration was set to 5 seconds in SETUP.conf – the start time for the mining process is synchronized (same conditions for each node).

[2] The local node it self is the fastest one, which has solved the "proof of work" task. The result will be propagated to all other nodes for verification.

[3] An other node has send a "proof of work" result. After successful verification by local node, a accept message will be propagated.



BUSINESS IMPACT

Secure and immutable Transaction Handling

Distributed storage, difficult to manipulate but easy to check. Perfect to store continuous transactions and make them available to all network-members. Secure and unbreakable.





Transparency, Tracking and Tracing

Dedicated operation mode to focus on tracing, tracing and data logging. Increase transparency and quality in your business applications based on rell.CHain.

Trust in Time

Automation of standard processes and operational processes through predefined audit-proof workflows. Lower costs and shorter process time through Smart Contracts on your own, fast and secure infrastructure.



