Secure audit of clinical registry records using Blockchain

Vedant Chauhan, Anthony Stell, and Richard Sinnott

University of Melbourne

e-Research New Zealand, 2019

Auckland, NZ

Overview

- Goal
- Background
- Blockchain
- Is Blockchain a good fit for Audit?
- Audit based on log files
- Results
  - Performance
  - Validation
- Audit based on database (MongoDB)
- Conclusion

Goal

Attackers are exploiting different vulnerabilities in the application to cause a security attack. One such vulnerability we are focusing on is ‘audit trail’.

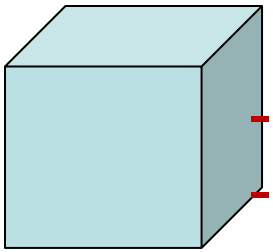
We propose a method of using Blockchain technology to establish the secure provenance of:

- Log-files taken from an online application using both flat ASCII text files (generated from the Apache log4j project).
- The same logs stored in database format (MongoDB).

Background

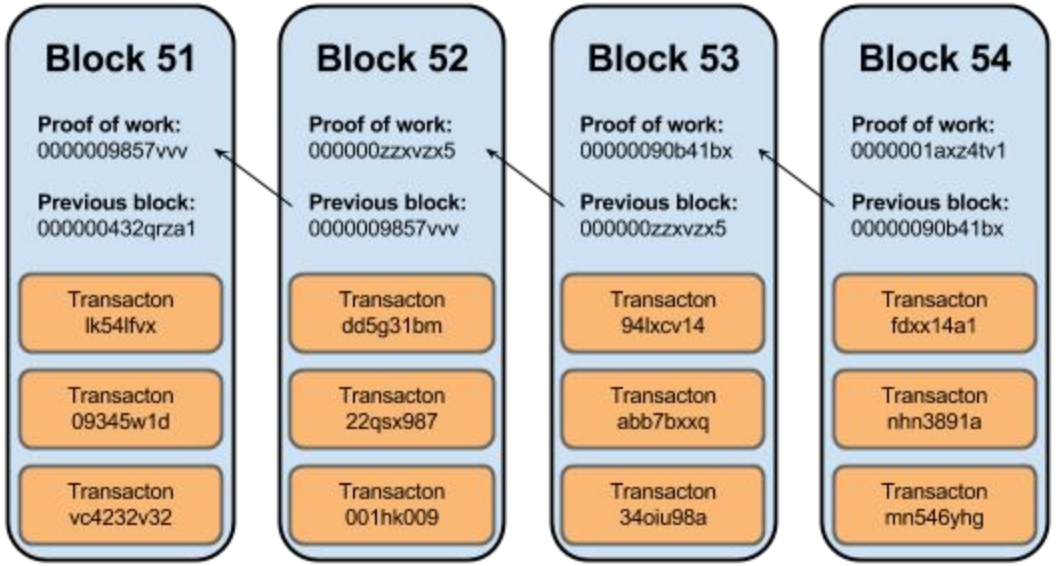
- What is audit in IT application?
  - Audit are used to document transactions
  - Audit are maintained in a log
- What is a log file?
  - Recording statements from code
  - Usage: Debugging
- Clinical registry: ENSAT
  - European Network for the Study of Adrenal Tumors
  - Generates a log daily using log4j

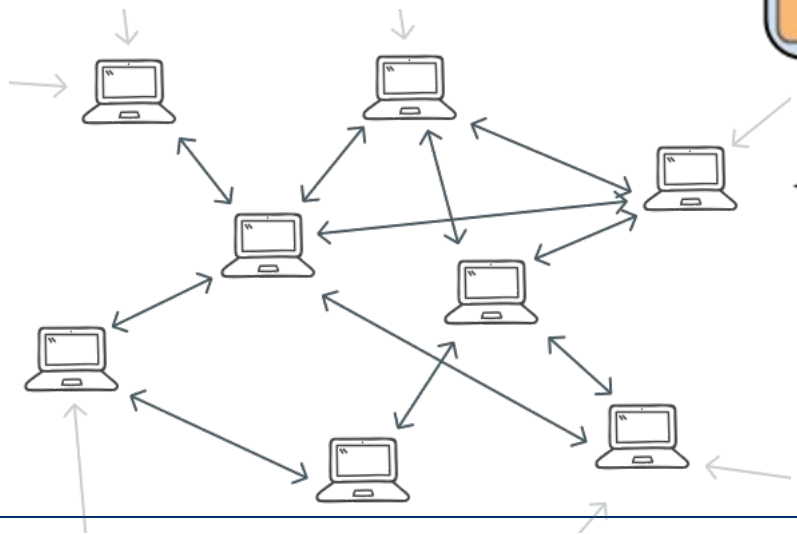
Blockchain



Block

-Hash
-Data
-Previous Hash
-Timestamp





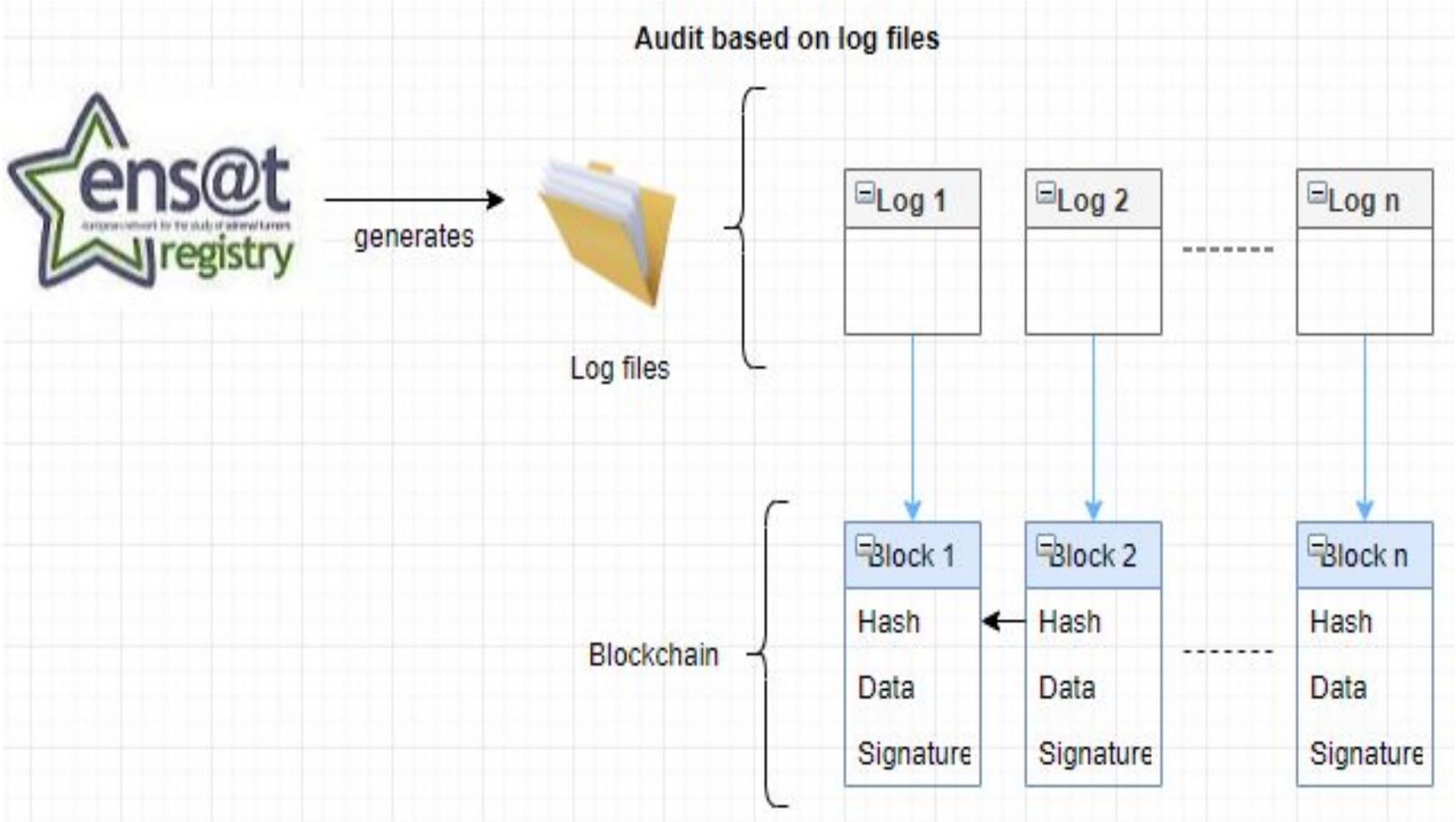
Blocks are “more secure” as you go further back in the chain

Is Blockchain a good fit for Audit?

- Opportunities:
  - Immutability and Distributed
  - Real-time verification
  - Coverage

- Challenges:
  - Security
  - No central authority

Audit based on log files



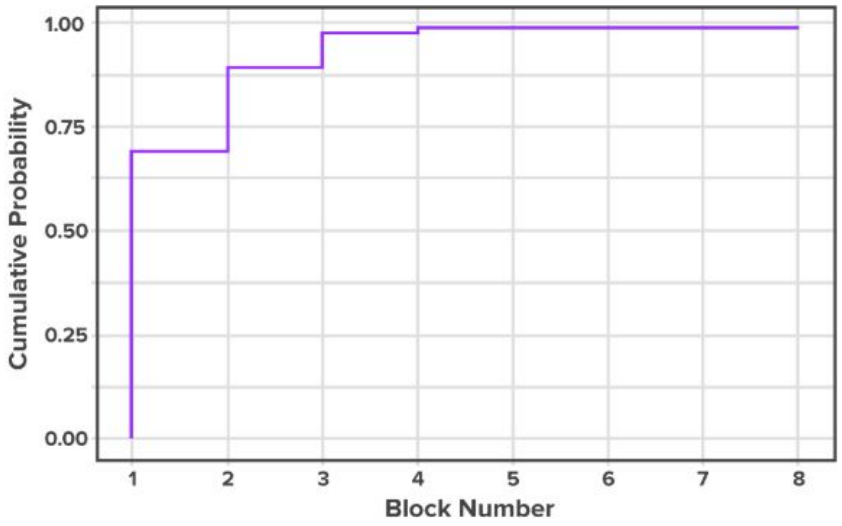
Results

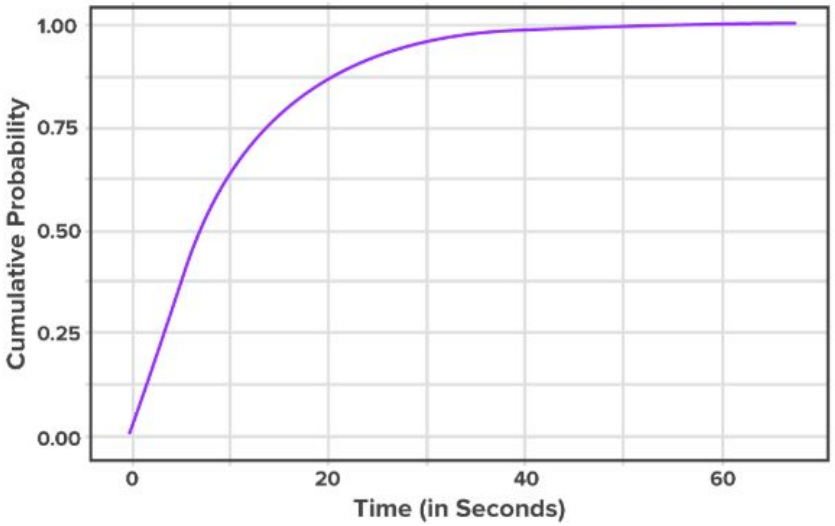
- Final structure

```
07 Feb 2019 19:59:38 rootLogger  - New session created (total active sessions = 1)
07 Feb 2019 20:00:06 rootLogger  - sign:[B@4366edcd
07 Feb 2019 20:00:08 rootLogger  - logChain:0f000853fa463637dd4b442079478d24619599d93336822beale18a2cba47cb95
07 Feb 2019 20:17:19 rootLogger  - Session destroyed (total active sessions = 0)
07 Feb 2019 20:19:19 rootLogger  - New session created (total active sessions = 1)
07 Feb 2019 20:26:08 rootLogger  - sign:[B@40429b83
07 Feb 2019 20:26:30 rootLogger  - ExportThreadPool destroyed...
07 Feb 2019 20:40:18 rootLogger  - New session created (total active sessions = 1)
07 Feb 2019 20:40:19 rootLogger  - sign:[B@7568b905
07 Feb 2019 20:42:12 rootLogger  - Session destroyed (total active sessions = 0)
07 Feb 2019 20:44:18 rootLogger  - ExportThreadPool destroyed...
07 Feb 2019 20:44:47 rootLogger  - New session created (total active sessions = 1)
07 Feb 2019 20:44:48 rootLogger  - sign:[B@5e01fbec
07 Feb 2019 20:46:18 rootLogger  - ExportThreadPool destroyed...
07 Feb 2019 20:52:22 rootLogger  - New session created (total active sessions = 1)
07 Feb 2019 20:52:23 rootLogger  - sign:[B@759cbc3f
07 Feb 2019 20:54:00 rootLogger  - ExportThreadPool destroyed...
07 Feb 2019 20:54:40 rootLogger  - New session created (total active sessions = 1)
07 Feb 2019 20:54:41 rootLogger  - sign:[B@4913a177
07 Feb 2019 20:54:54 rootLogger  - New session created (total active sessions = 1)
07 Feb 2019 20:54:55 rootLogger  - sign:[B@315a9de0
07 Feb 2019 20:55:25 rootLogger  - Session destroyed (total active sessions = 0)
07 Feb 2019 20:56:37 rootLogger  - Session destroyed (total active sessions = 0)
07 Feb 2019 20:59:00 security.UserCheck  - Running user check (login request)...
07 Feb 2019 20:59:00 rootLogger  - emailUsername is: test@testuser.com
07 Feb 2019 20:59:05 security.UserCheck  - accountActive: true
07 Feb 2019 20:59:05 security.UserCheck  - membershipCurrent: true
07 Feb 2019 20:59:05 security.UserCheck  - Login credentials checked against SHA-256 (test@testuser.com)...
07 Feb 2019 20:59:05 security.UserCheck  - hashedInput: 0761095ac9a118ab40faf8badf59034baa7fedf665d035e0edfc6dcaeca76d56
07 Feb 2019 20:59:05 security.UserCheck  - Out of main checking clause...
07 Feb 2019 20:59:05 security.UserCheck  - Into other flag checks...
07 Feb 2019 20:59:05 security.UserCheck  - responseFlag (UserCheck.java): 0
07 Feb 2019 20:59:05 rootLogger  - responseFlag after checkUserDetails: 0
07 Feb 2019 20:59:05 security.UserCheck  - === User 'test@testuser.com' successfully logged in at 20:54 ===
07 Feb 2019 20:59:05 ConnectBean.ConnectBean  - Main database connection assigned from DataSource pool...
07 Feb 2019 20:59:05 ConnectBean.ConnectBean  - Security database connection assigned from DataSource pool...
07 Feb 2019 20:59:05 ConnectBean.ConnectBean  - Parameter database connection assigned from DataSource pool...
07 Feb 2019 20:59:05 ConnectBean.ConnectBean  - Center callout database connection assigned from DataSource pool...
```

Results: Performance

- Avg. Block Time

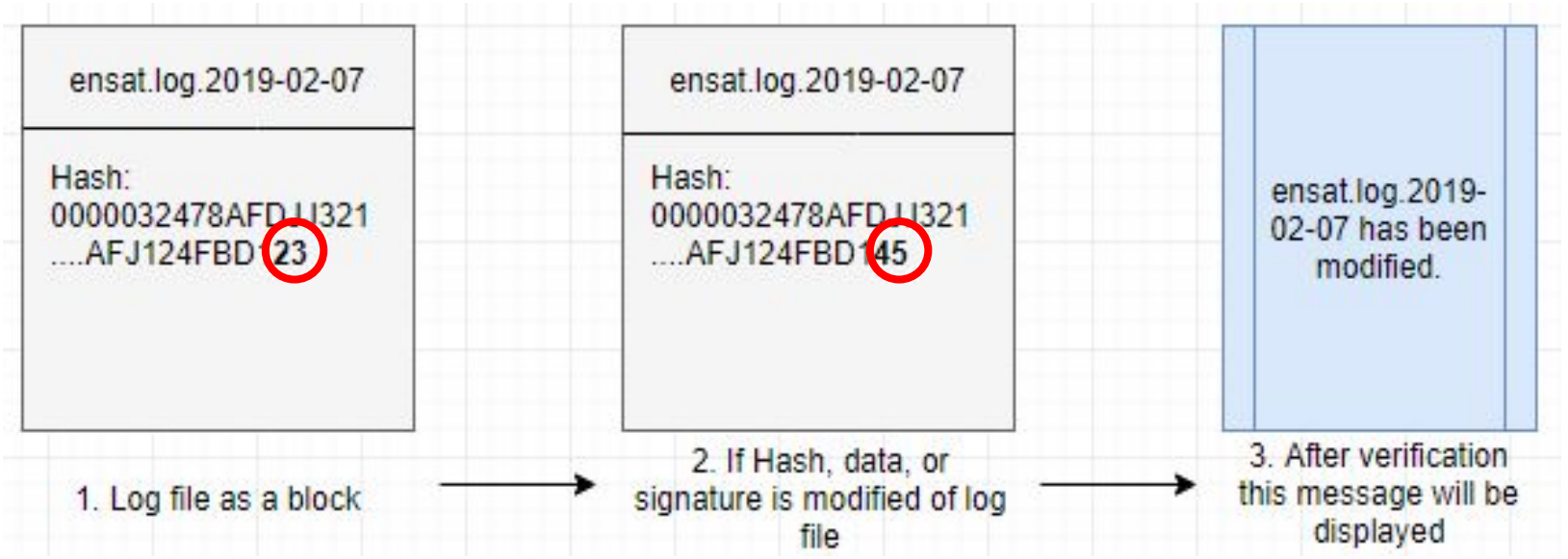




- Avg. time taken to generate a block is 5 to 10 seconds.
- Faster than public blockchains.

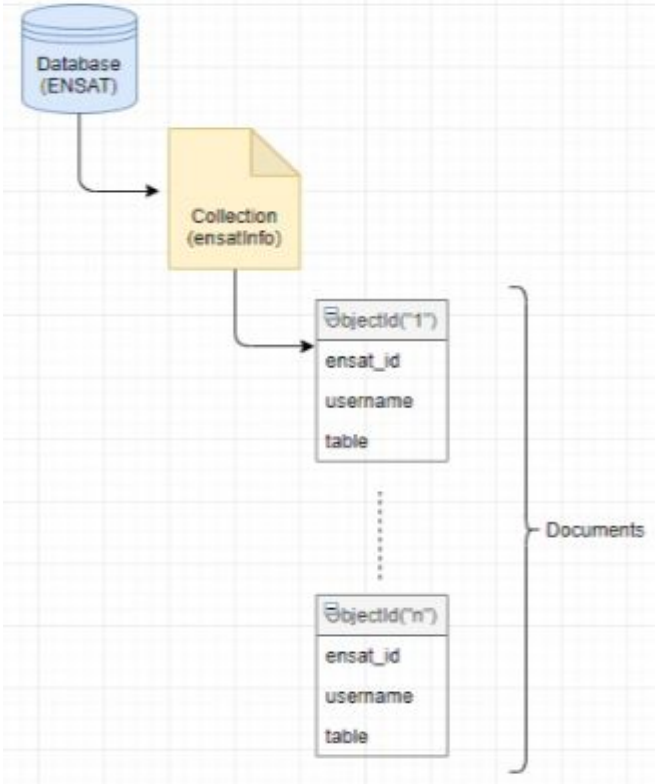
Results: Validation

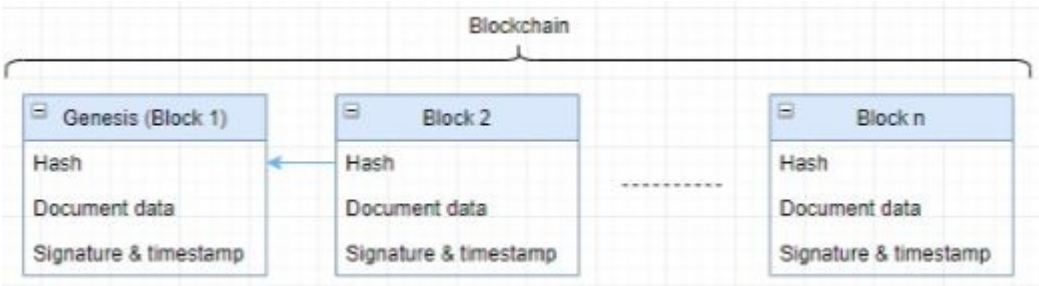
- Verifying the hashes:
  - From log files and the blockchain
  - Any changes to file without authorization is notified.



Audit based on database (MongoDB)

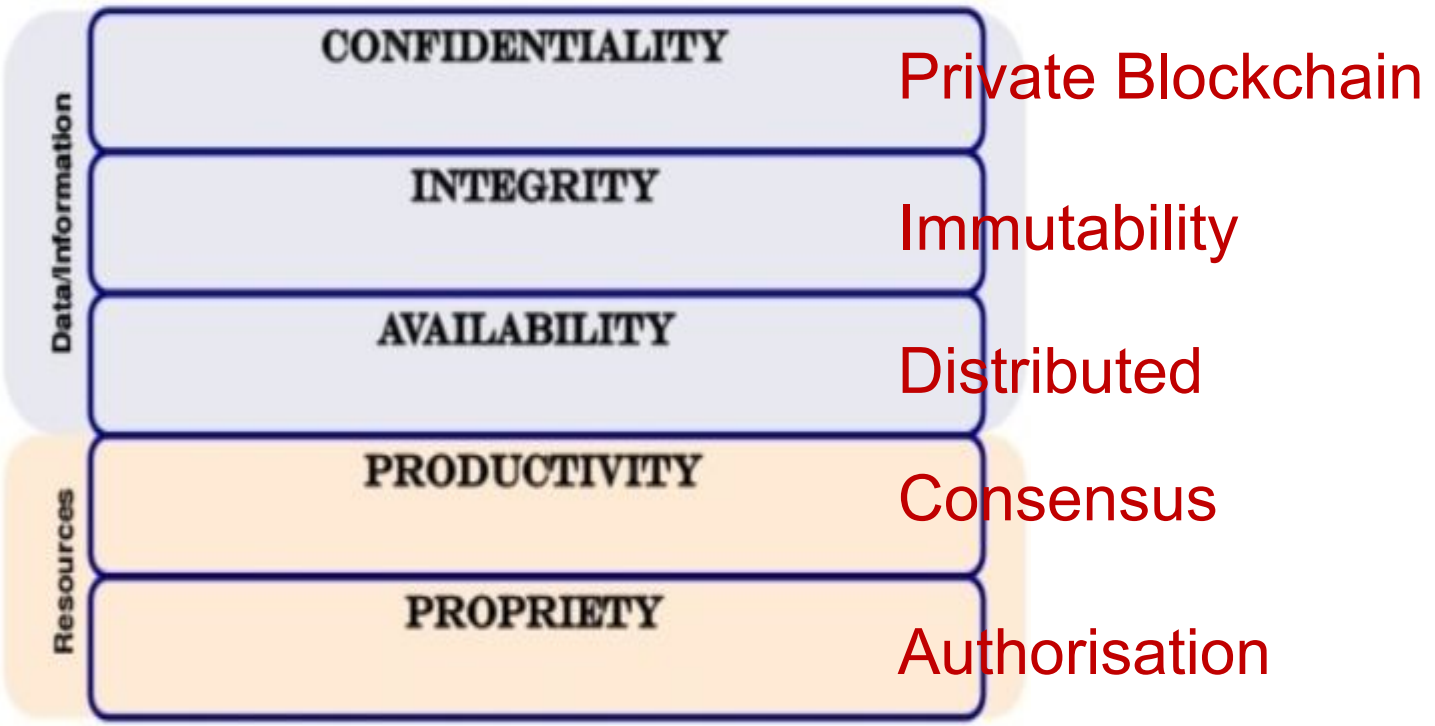
- MongoDB properties:
  - NoSQL Database
  - Unstructured
  - JSON-oriented
- Flow of the program:





Conclusion

- Proposed system provides a security model satisfying:



References

- "ENS@T - European Network for the Study of Adrenal Tumors", *Registry.ensat.org*, 2019. [Online]. Available: https://registry.ensat.org/.
- Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).

Acknowledgements

- Special thanks to my co-authors from Melbourne eResearch Group

THE UNIVERSITY OF
MELBOURNE