www.chainfrog.com

chainfrog

# CHECKLIST

Find us at www.chainfrog.com

# Could you just use a database?

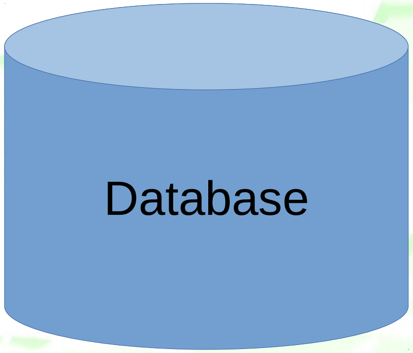
- If your current and future project requirements can be satisfied by a relational database:

  don’t use a blockchain

- Databases have decades of:

  - Development
  - Debugging
  - Optimization

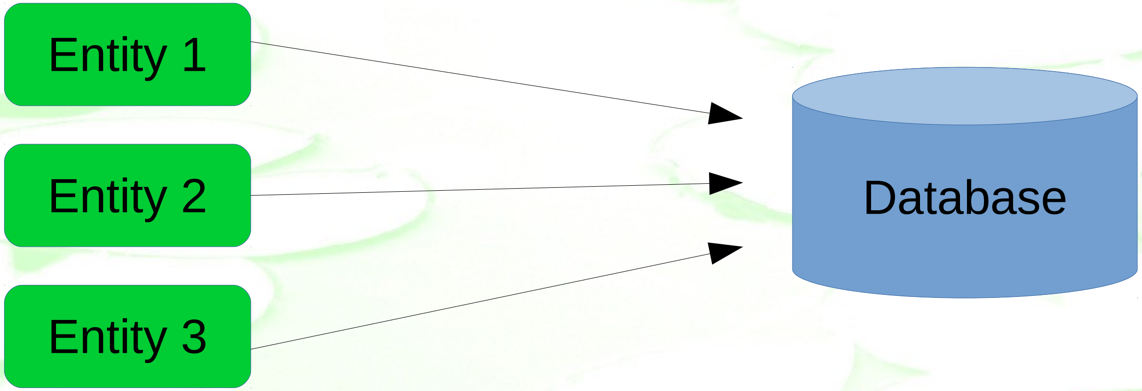
  backing them up...



# Now lets move on to when to use a blockchain

# Do you have multiple participants?

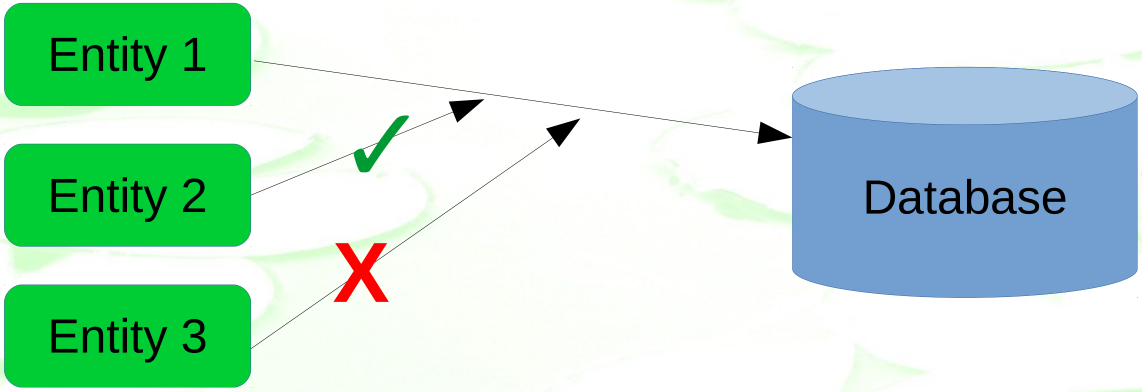
- Blockchains are a way of allowing different parties to generate transactions or data records



- For example, an invoice system, with different entities charging, and different parties making payments

# Do the multiple participants not trust each other fully?

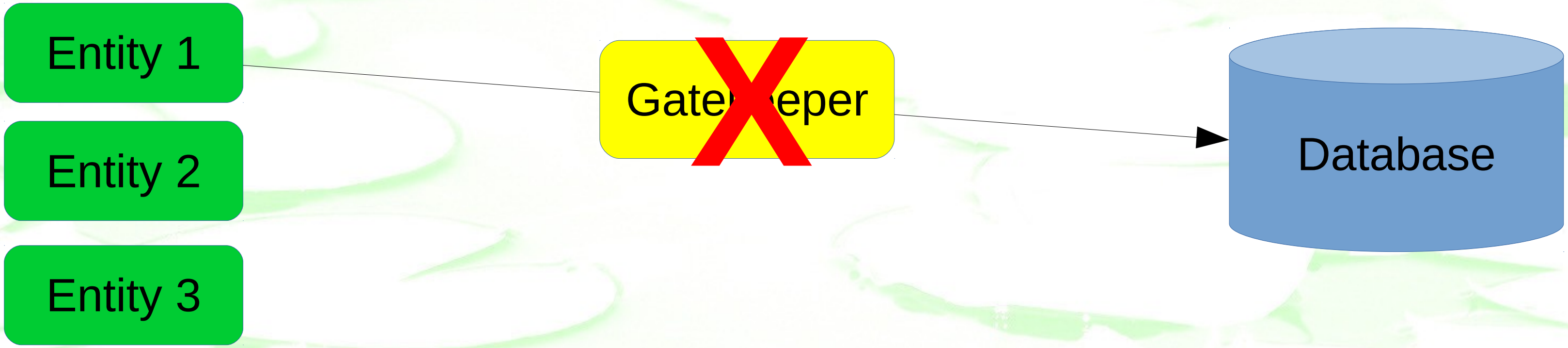
- Blockchains are a way of building trust between separate entities (should something be added to the database or not?)



- For example, the different entities may have a common goal (ensuring payments to their members), but may have different economic incentives, e.g. they may be based in different countries

# Is there no universally trusted third-party?

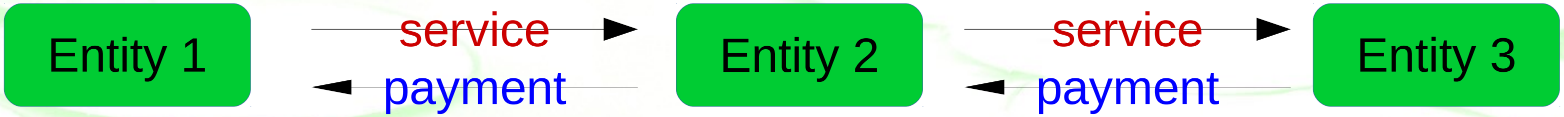
- Blockchains avoid the need for an intermediary



- If there is a suitable third-party that can be trusted to vet every database transaction then a blockchain is not needed.
- However, such gatekeepers usually charge for their service (e.g. Visa taking 2.5%)

# Are the database transactions dependent on each other?

- Blockchains are great if the transactions depend on each other or are needed to validate each other



- Blockchains overcome “double-spend”
- They timestamp reports, providing an immutable chronology to records
- They provide an authoritative finalized log of these records and transactions

# Added bonus: native assets

- Because they are tamper-proof, distributed, and consensus based “ledgers”, it is trivial to implement “native assets” on a blockchain:
  - Tokens,
  - Credits,
  - Crypto-currencies,
  - Tickets,
  - Votes,
  - Shares
- You decide how they are issued, who initially gets them, and under what condition they are transferred

# Note of caution: forks!

- A known problem with blockchains is “the fork”
- Typically caused by changes in the consensus protocol, or other underlying rules of the blockchain, it causes one group to follow one branch of the chain, and the other group to follow another
- Unlike Android or web apps, you can’t make major architectural changes to your blockchain and “push it out there” without getting your user base to agree