Some slides glued together for the cryptology seminar 2008

Dan Bogdanov
University of Tartu / Cybernetica

Motivation

Protecting personal data

Protecting personal data

- Personal data includes medical and financial records, beliefs and preferences

Protecting personal data

- Personal data includes medical and financial records, beliefs and preferences

- This information should not become public

Protecting personal data

- Personal data includes medical and financial records, beliefs and preferences

- This information should not become public

- Every database containing such values in an identifiable form is a risk to our privacy

Protecting personal data

- Personal data includes medical and financial records, beliefs and preferences

- This information should not become public

- Every database containing such values in an identifiable form is a risk to our privacy

- Our goal is to build a better database

State of the art

State of the art

- Information systems generally do not preserve privacy

State of the art

- Information systems generally do not preserve privacy
- The standard solution is to password-protect the data

State of the art

- Information systems generally do not preserve privacy
- The standard solution is to password-protect the data
  - The data analyst will still see all the details

State of the art

- Information systems generally do not preserve privacy
- The standard solution is to password-protect the data
  - The data analyst will still see all the details
- Another idea is to introduce errors to the dataset

State of the art

- Information systems generally do not preserve privacy
- The standard solution is to password-protect the data
  - The data analyst will still see all the details
- Another idea is to introduce errors to the dataset
  - The privacy provided by this method is not provable

State of the art

- Information systems generally do not preserve privacy
- The standard solution is to password-protect the data
  - The data analyst will still see all the details
- Another idea is to introduce errors to the dataset
  - The privacy provided by this method is not provable
- Cryptographic techniques are looking promising

Share computing

Share computing

- Secret sharing is a method for distributing a secret value between several participants

Share computing

- Secret sharing is a method for distributing a secret value between several participants
    - The original value cannot be constructed without access to all the pieces (called *shares*)

Share computing

- Secret sharing is a method for distributing a secret value between several participants
    - The original value cannot be constructed without access to all the pieces (called *shares*)
- In share computing, the processed data is stored in a secret-shared form

Share computing

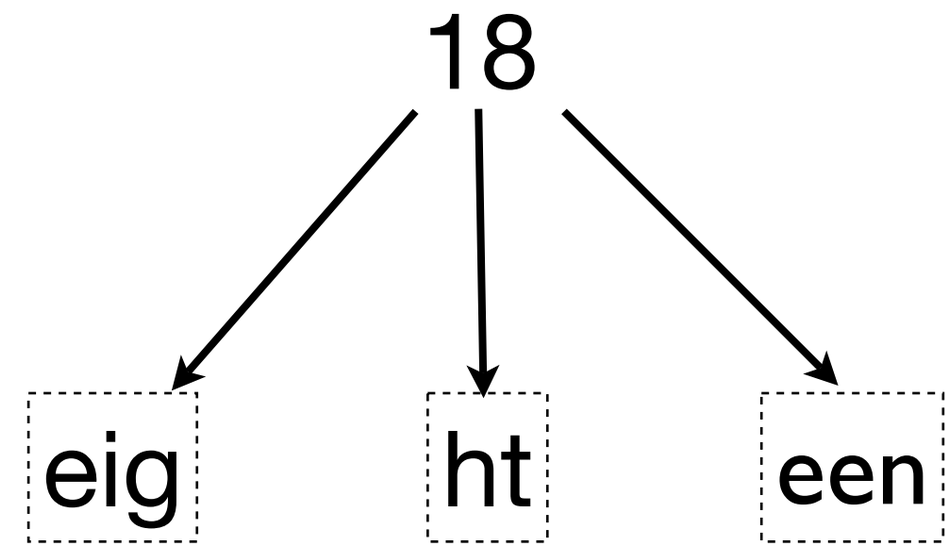
- Secret sharing is a method for distributing a secret value between several participants
  - The original value cannot be constructed without access to all the pieces (called *shares*)
- In share computing, the processed data is stored in a secret-shared form
- Multi-party computation allows us to process shares

Intuition to secret sharing

the input secret 18

Intuition to secret sharing

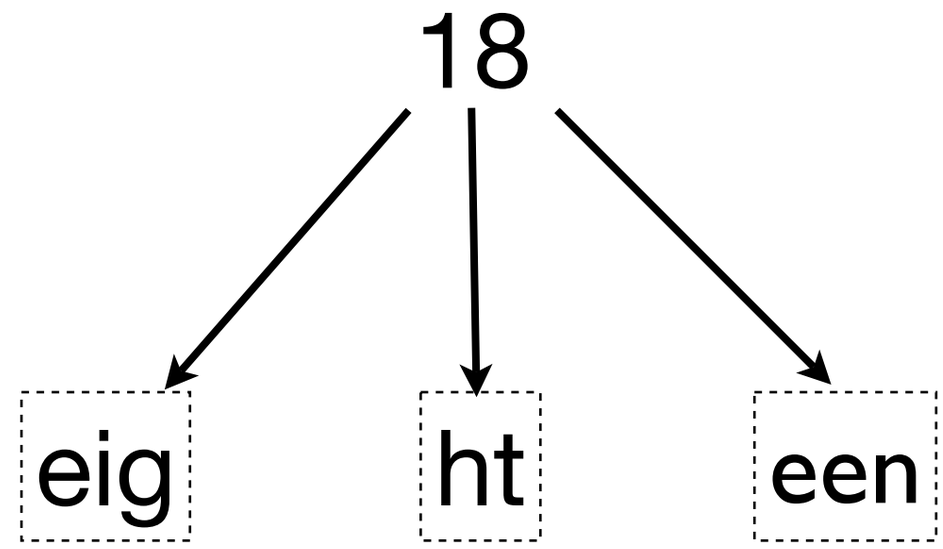
the input secret



shares

Intuition to secret sharing

the input secret



shares

reconstructed value

Not all schemes are good

Not all schemes are good

- Assume that we have secret-shared 18 and 103

Not all schemes are good

- Assume that we have secret-shared 18 and 103
- The shown scheme is not good for computing

Not all schemes are good

- Assume that we have secret-shared 18 and 103
- The shown scheme is not good for computing

eig ht een

Not all schemes are good

- Assume that we have secret-shared 18 and 103
- The shown scheme is not good for computing

	eig	ht	een
+	hundred	and	three

Not all schemes are good

- Assume that we have secret-shared 18 and 103
- The shown scheme is not good for computing

	eig	ht	een
+	hundred	and	three
		?	

Not all schemes are good

- Assume that we have secret-shared 18 and 103
- The shown scheme is not good for computing

	eig	ht	een
+	hundred	and	three

Not all schemes are good

- Assume that we have secret-shared 18 and 103
- The shown scheme is not good for computing

eig	ht	een
+ hundred	and	three
eighundred		

Not all schemes are good

- Assume that we have secret-shared 18 and 103
- The shown scheme is not good for computing

	eig	ht	een
+	hundred	and	three
	eighundred	htand	

Not all schemes are good

- Assume that we have secret-shared 18 and 103
- The shown scheme is not good for computing

eig	ht	een
+ hundred	and	three
eighundred	htand	eenthree

Homomorphic schemes are good

Homomorphic schemes are good

- The additive scheme is homomorphic

Homomorphic schemes are good

- The additive scheme is homomorphic
- $18 = 7 + 2 + 9$ and $103 = 24 + 50 + 29$

Homomorphic schemes are good

- The additive scheme is homomorphic
- 18 = 7 + 2 + 9 and 103 = 24 + 50 + 29

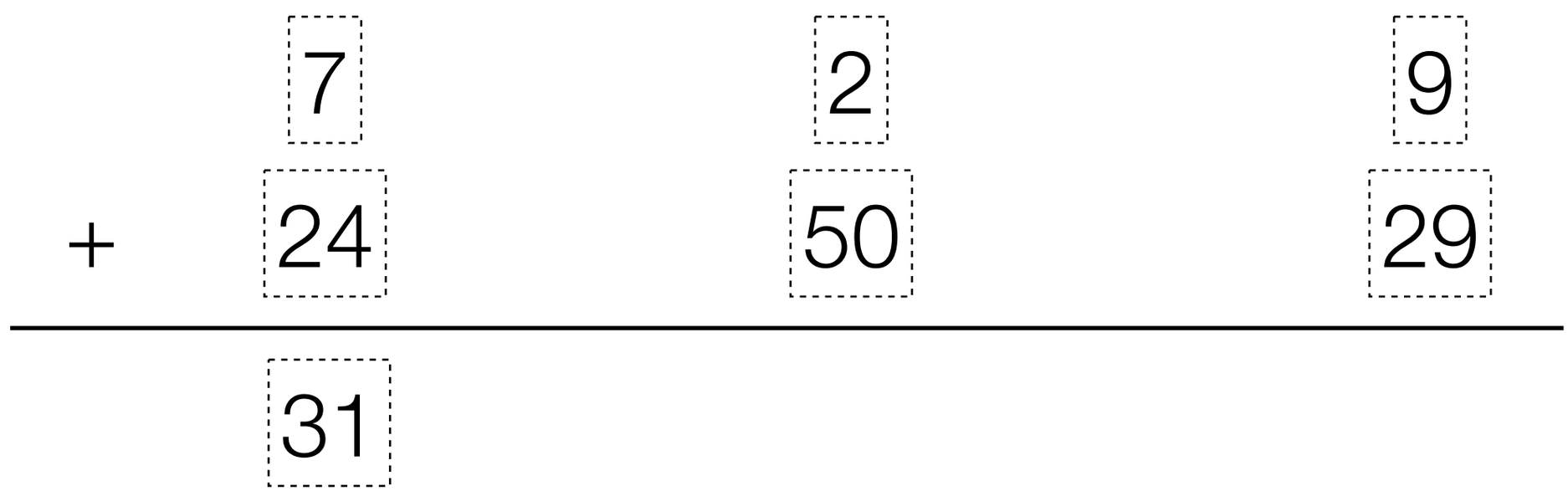
Homomorphic schemes are good

- The additive scheme is homomorphic
- 18 = 7 + 2 + 9 and 103 = 24 + 50 + 29

	7	2	9
+	24	50	29

Homomorphic schemes are good

- The additive scheme is homomorphic
- 18 = 7 + 2 + 9 and 103 = 24 + 50 + 29



Homomorphic schemes are good

- The additive scheme is homomorphic
- 18 = 7 + 2 + 9 and 103 = 24 + 50 + 29

	7	2	9
+	24	50	29
	31	52	

Homomorphic schemes are good

- The additive scheme is homomorphic
- 18 = 7 + 2 + 9 and 103 = 24 + 50 + 29

	7	2	9
+	24	50	29
	31	52	38

Homomorphic schemes are good

- The additive scheme is homomorphic
- 18 = 7 + 2 + 9 and 103 = 24 + 50 + 29

	7		2		9		
+	24		50		29		
	31	+	52	+	38		= 121

Introducing Sharemind

A privacy-preserving computer

A privacy-preserving computer

- Sharemind is a distributed virtual processor

A privacy-preserving computer

- Sharemind is a distributed virtual processor
- The processor performs share computing

A privacy-preserving computer

- Sharemind is a distributed virtual processor
- The processor performs share computing
  - the data is stored using the additive scheme

A privacy-preserving computer

- Sharemind is a distributed virtual processor
- The processor performs share computing
  - the data is stored using the additive scheme
  - multi-party computation protocols are applied

A privacy-preserving computer

- Sharemind is a distributed virtual processor
- The processor performs share computing
  - the data is stored using the additive scheme
  - multi-party computation protocols are applied
- It is information-theoretically secure in a semi-honest model with three parties

Our innovation

Our innovation

- The main design goal of Sharemind is performance

Our innovation

- The main design goal of Sharemind is performance
    - Sharemind performs best with larger input vectors and single operations are relatively slower

Our innovation

- The main design goal of Sharemind is performance
  - Sharemind performs best with larger input vectors and single operations are relatively slower
  - This is due to extensive vectorisation

Our innovation

- The main design goal of Sharemind is performance
    - Sharemind performs best with larger input vectors and single operations are relatively slower
        - This is due to extensive vectorisation
- We also want it to be practically usable

Our innovation

- The main design goal of Sharemind is performance
  - Sharemind performs best with larger input vectors and single operations are relatively slower
    - This is due to extensive vectorisation
- We also want it to be practically usable
  - Writing Sharemind applications is relatively easy

What can it compute?

What can it compute?

- Sharemind can securely:

What can it compute?

- Sharemind can securely:
  - add or multiply two values

What can it compute?

- Sharemind can securely:
  - add or multiply two values
  - multiply a value by a constant

What can it compute?

- Sharemind can securely:
  - add or multiply two values
  - multiply a value by a constant
  - extract bits from a value

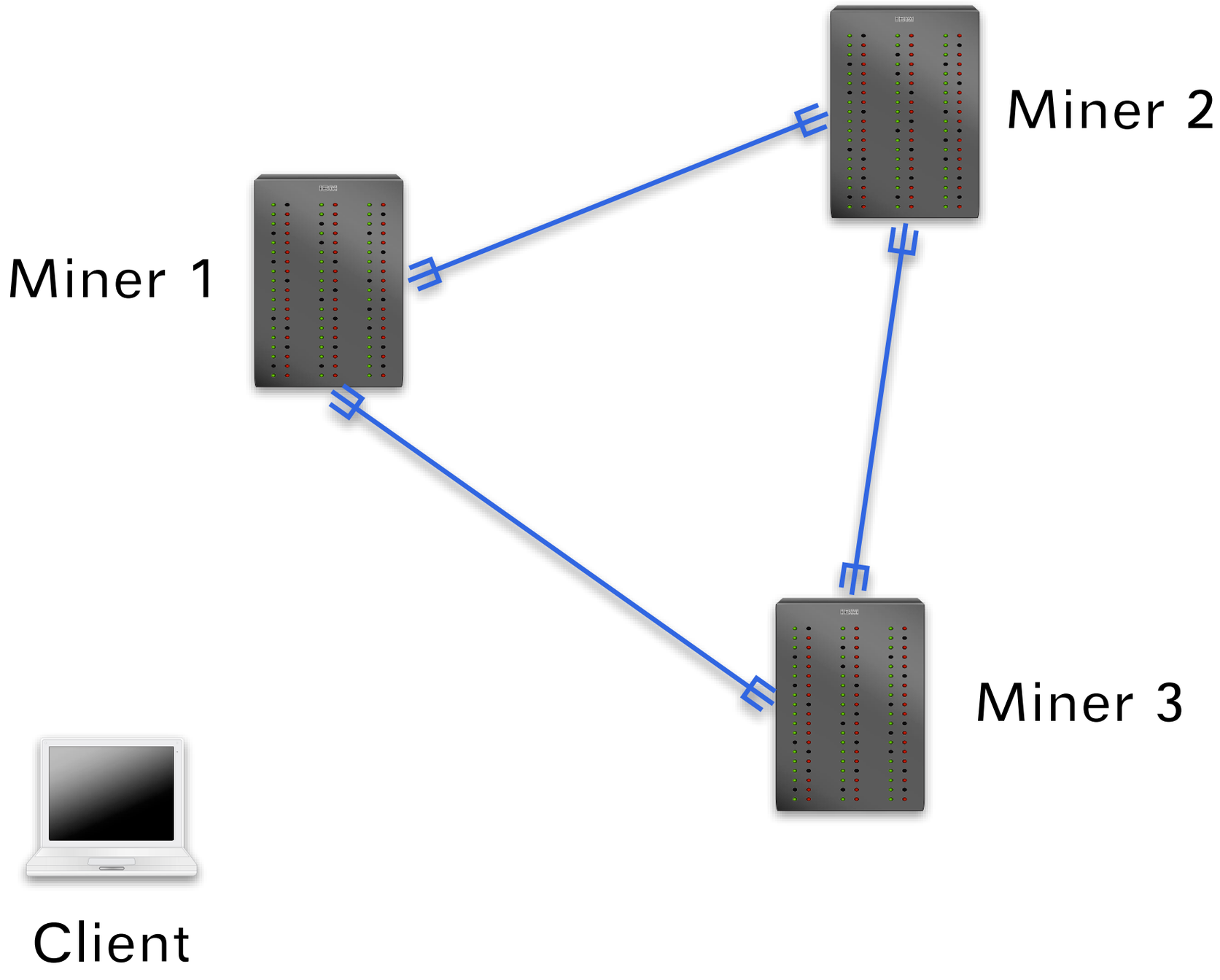
What can it compute?

- Sharemind can securely:
  - add or multiply two values
  - multiply a value by a constant
  - extract bits from a value
  - determine if two values are equal

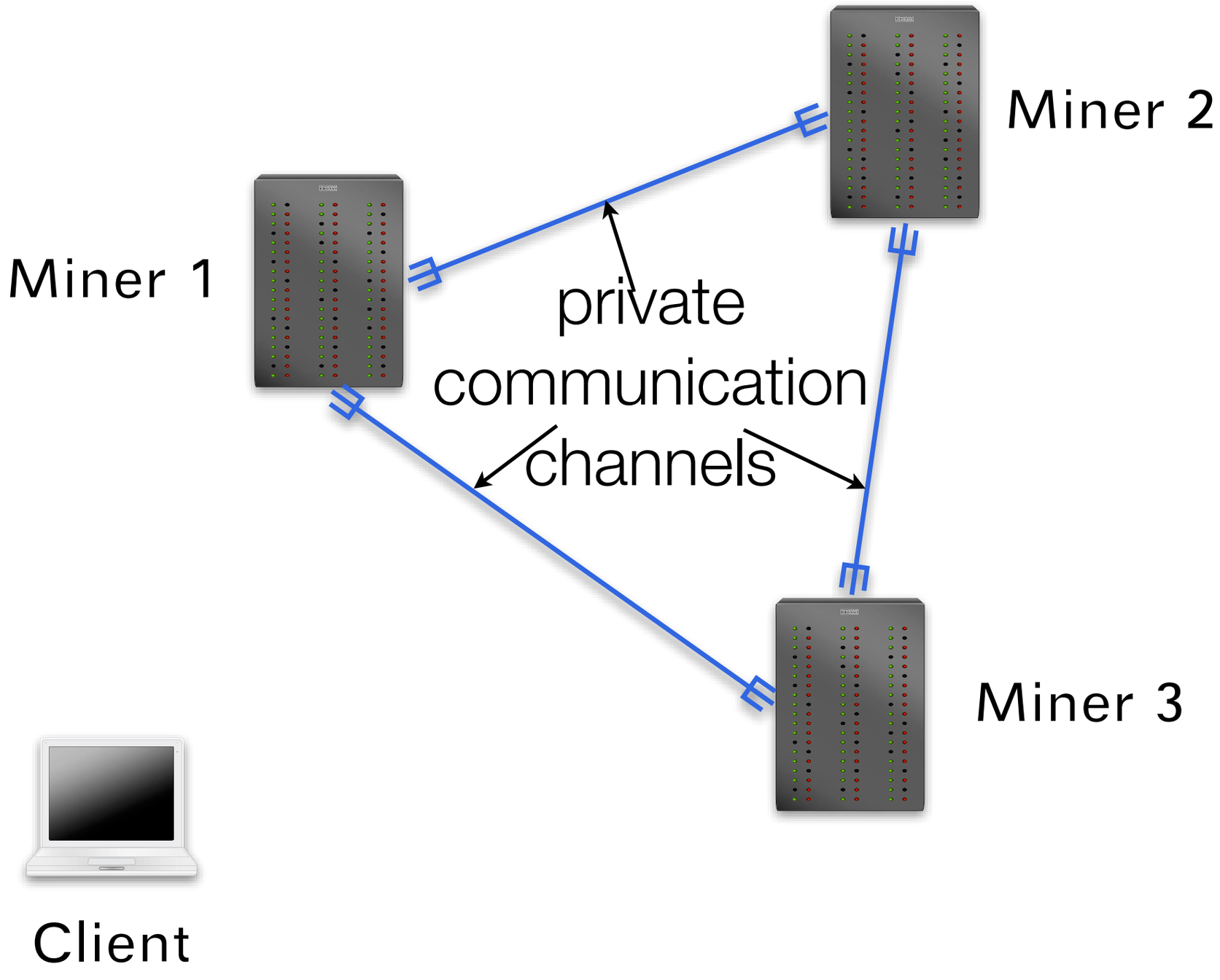
What can it compute?

- Sharemind can securely:
  - add or multiply two values
  - multiply a value by a constant
  - extract bits from a value
  - determine if two values are equal
  - determine the greater one of two values

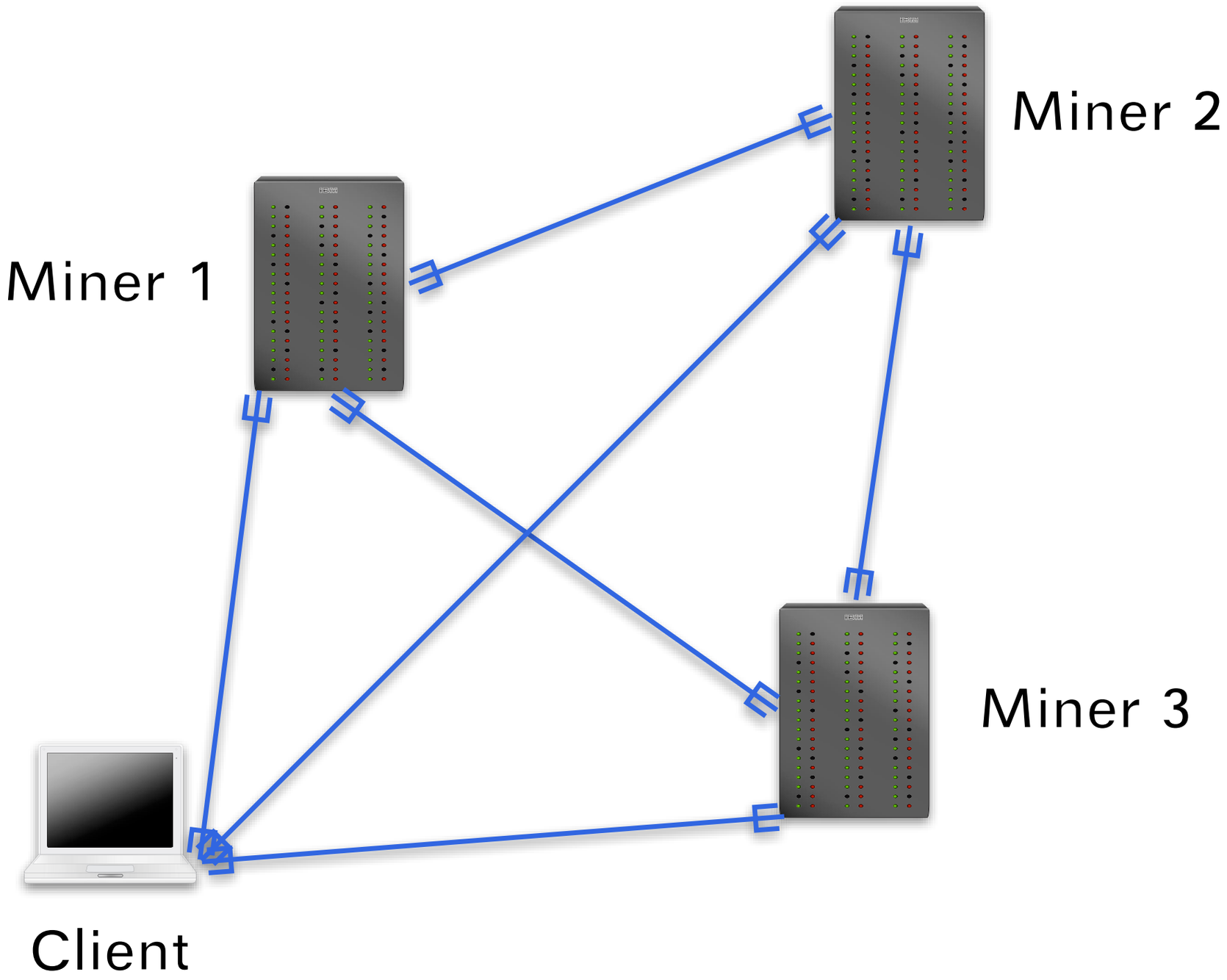
What does it look like?



What does it look like?



Clients connect to miners

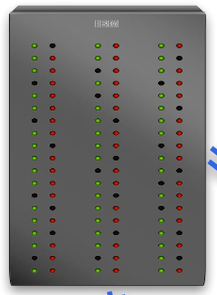


Executing instructions

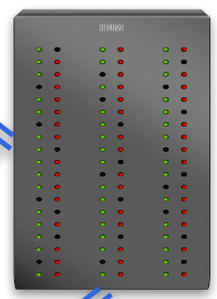
Client

Virtual processor

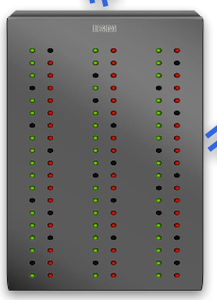
Miner 1



Miner 2

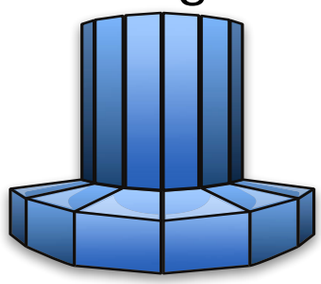


Miner 3

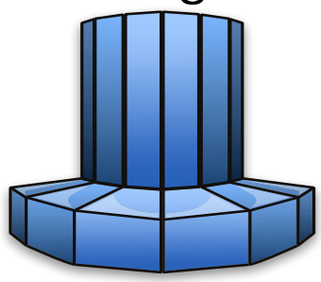


Storage

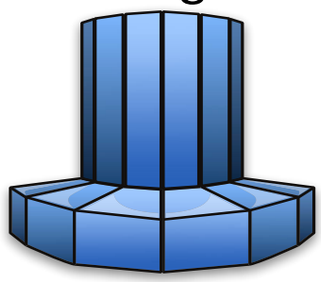
Storage 1



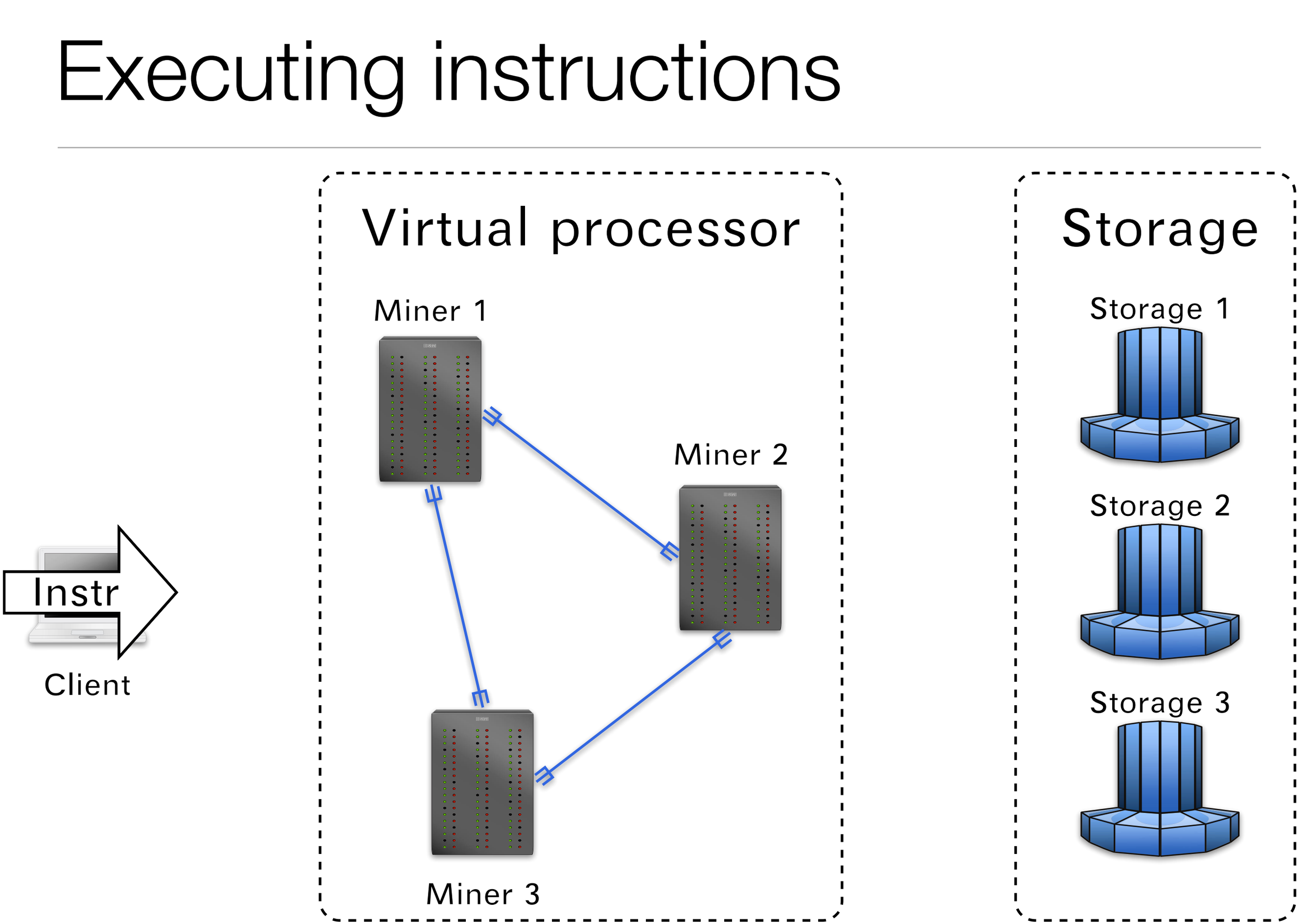
Storage 2



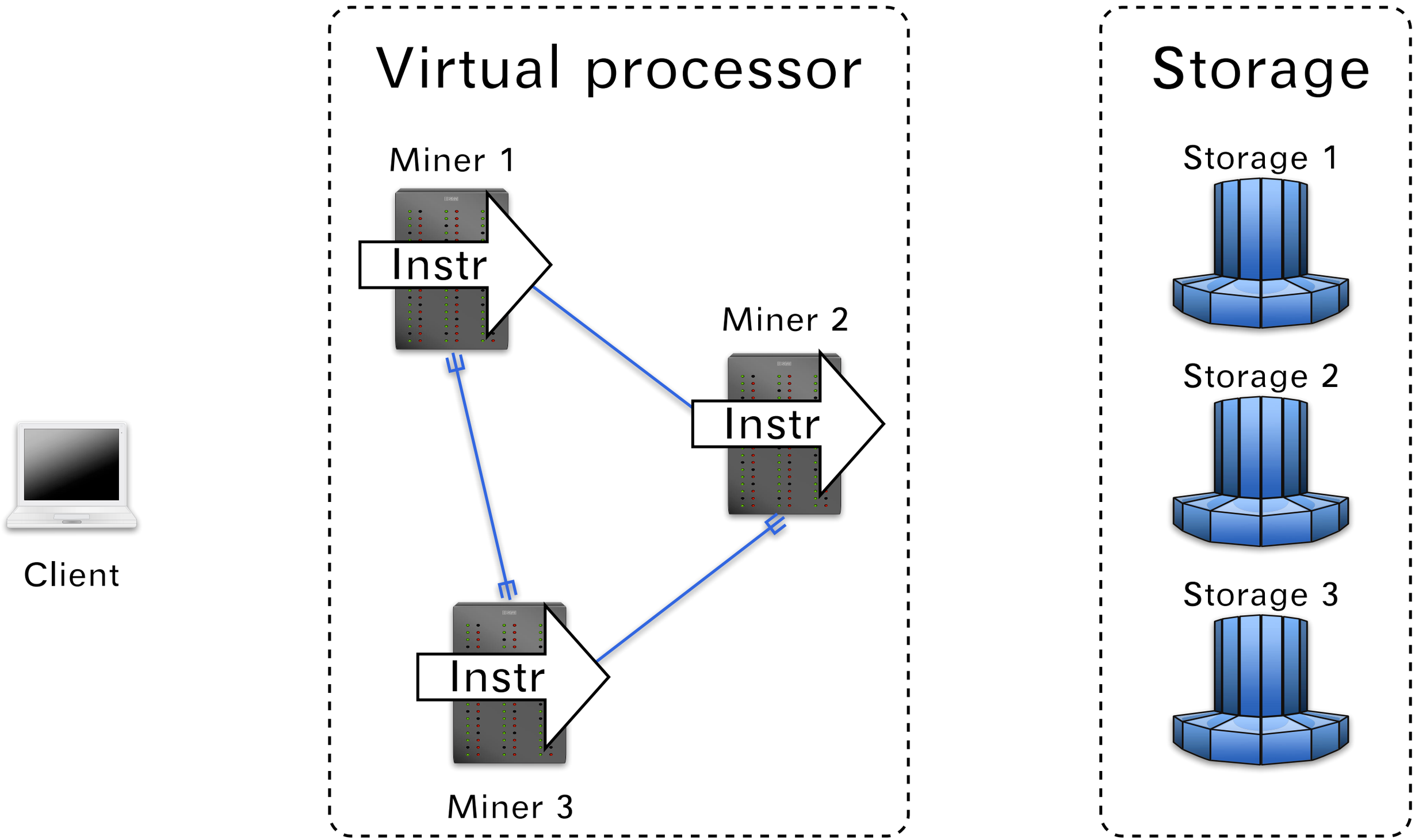
Storage 3



Executing instructions



Executing instructions

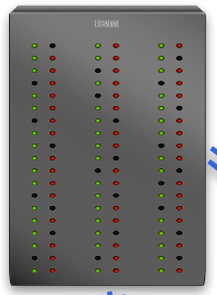


Executing instructions

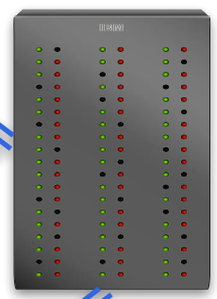
Client

Virtual processor

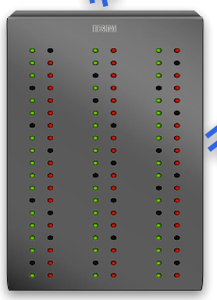
Miner 1



Miner 2

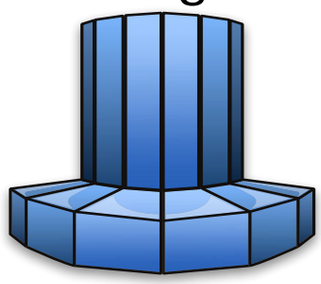


Miner 3

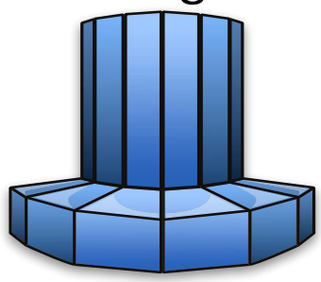


Storage

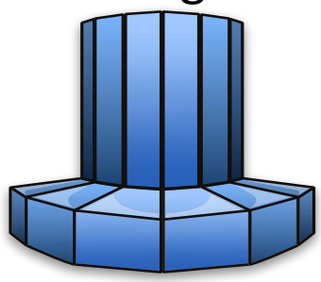
Storage 1



Storage 2



Storage 3

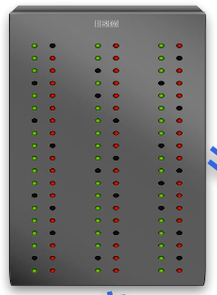


Executing instructions

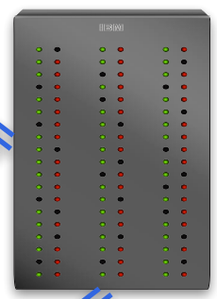
Client

Virtual processor

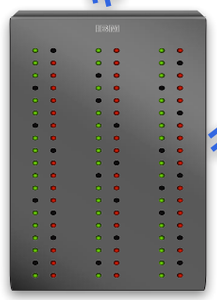
Miner 1



Miner 2

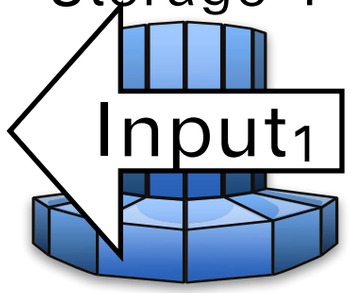


Miner 3

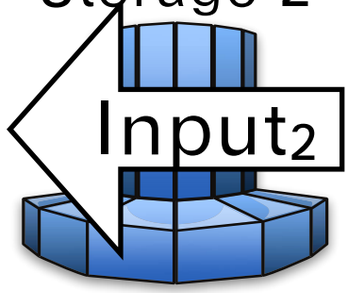


Storage

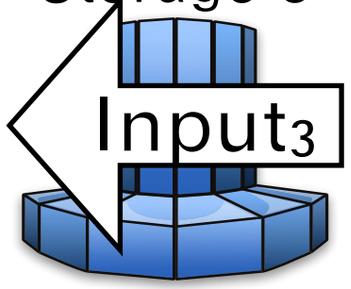
Storage 1



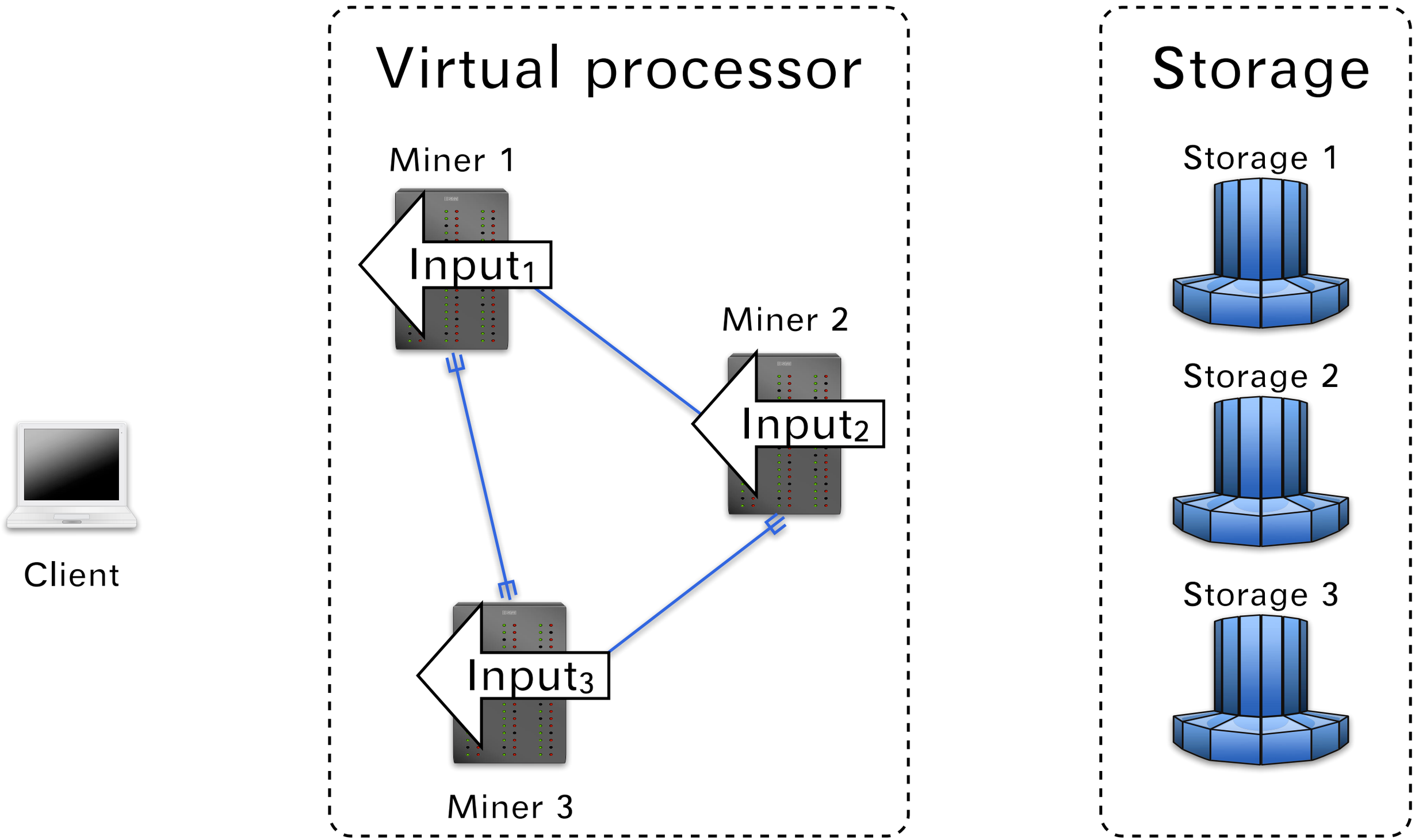
Storage 2



Storage 3



Executing instructions

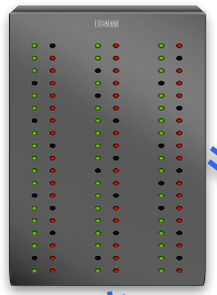


Executing instructions

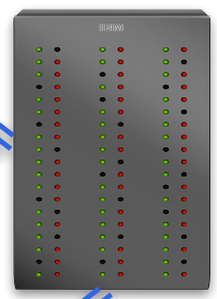
Client

Virtual processor

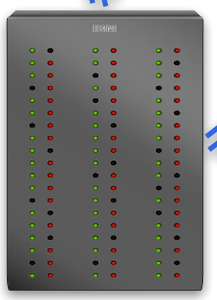
Miner 1



Miner 2

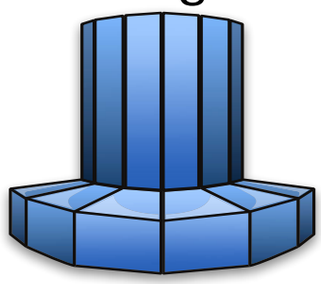


Miner 3

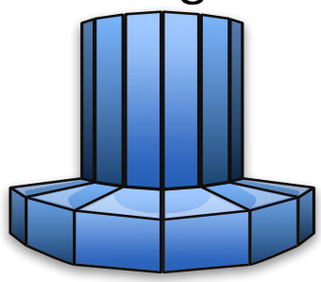


Storage

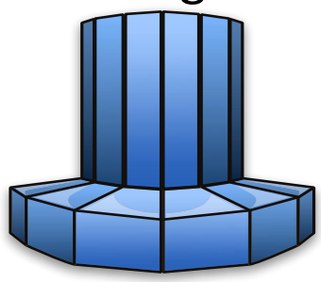
Storage 1



Storage 2



Storage 3

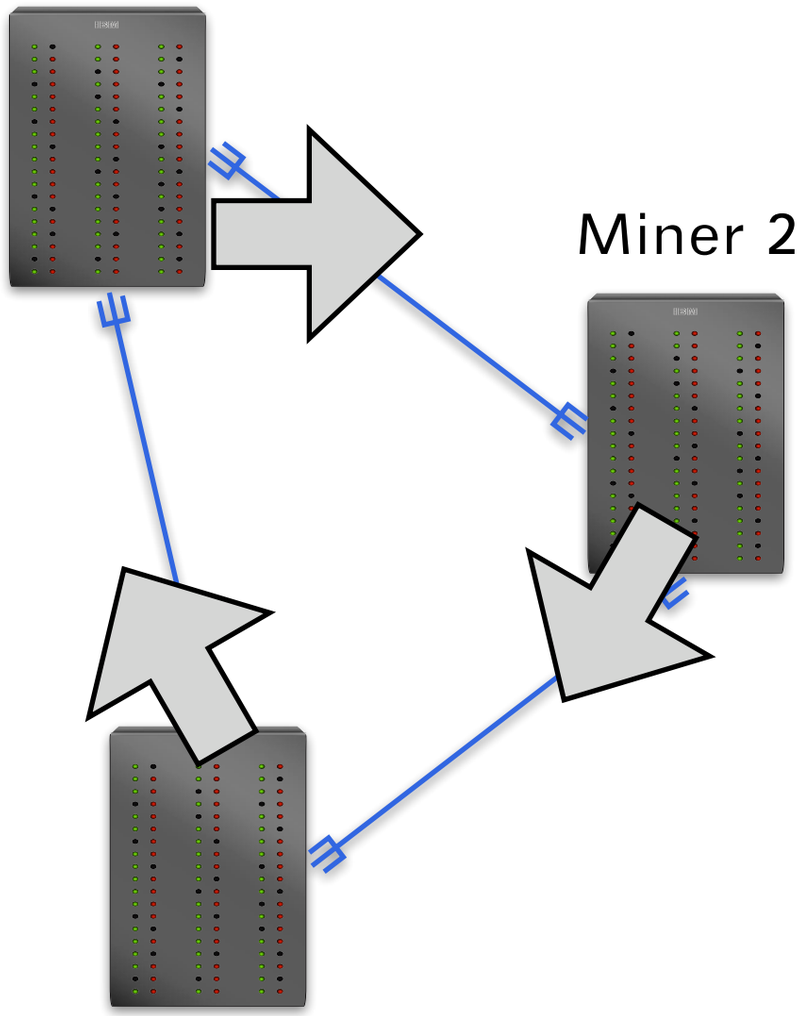


Executing instructions

Client

Virtual processor

Miner 1

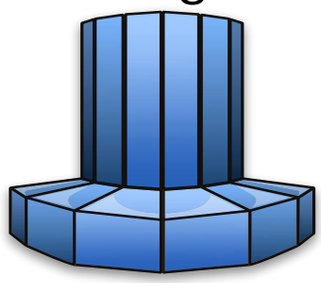


Miner 2

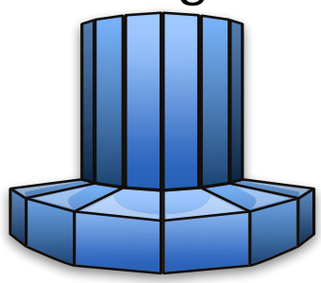
Miner 3

Storage

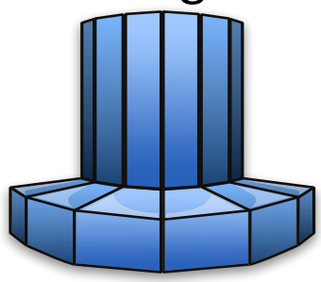
Storage 1



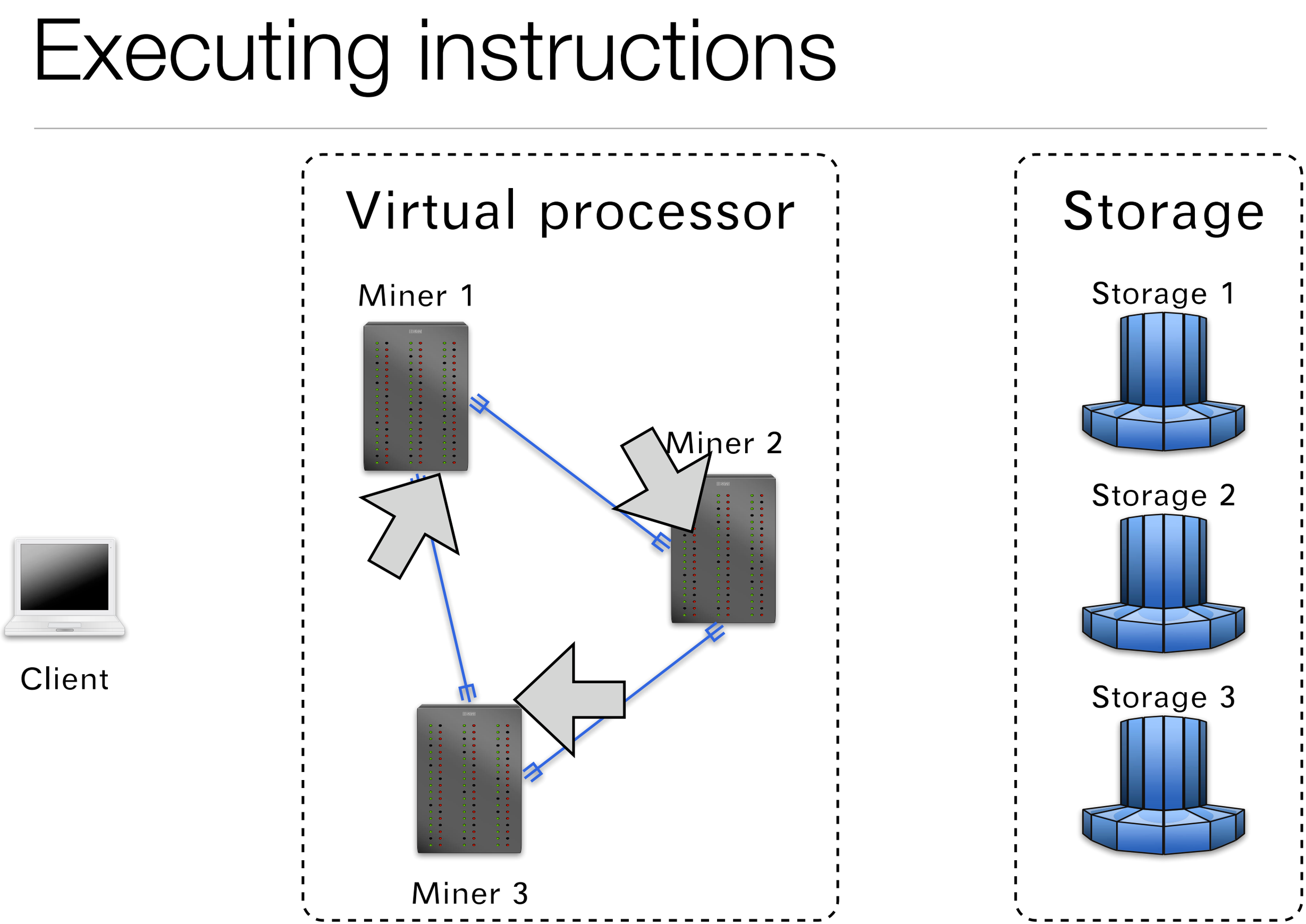
Storage 2



Storage 3



Executing instructions

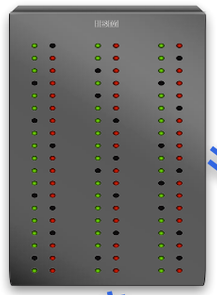


Executing instructions

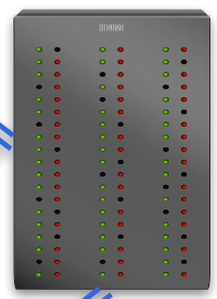
Client

Virtual processor

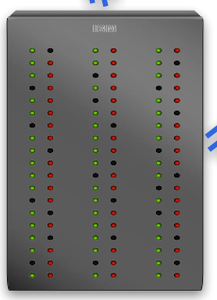
Miner 1



Miner 2

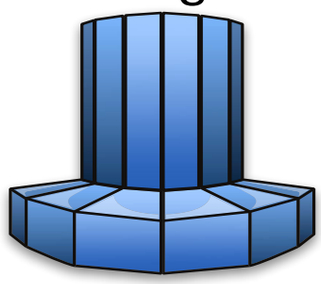


Miner 3

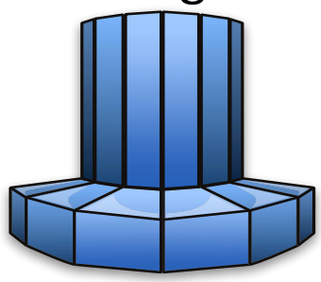


Storage

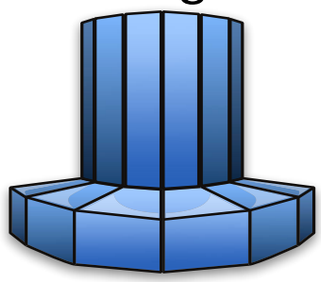
Storage 1



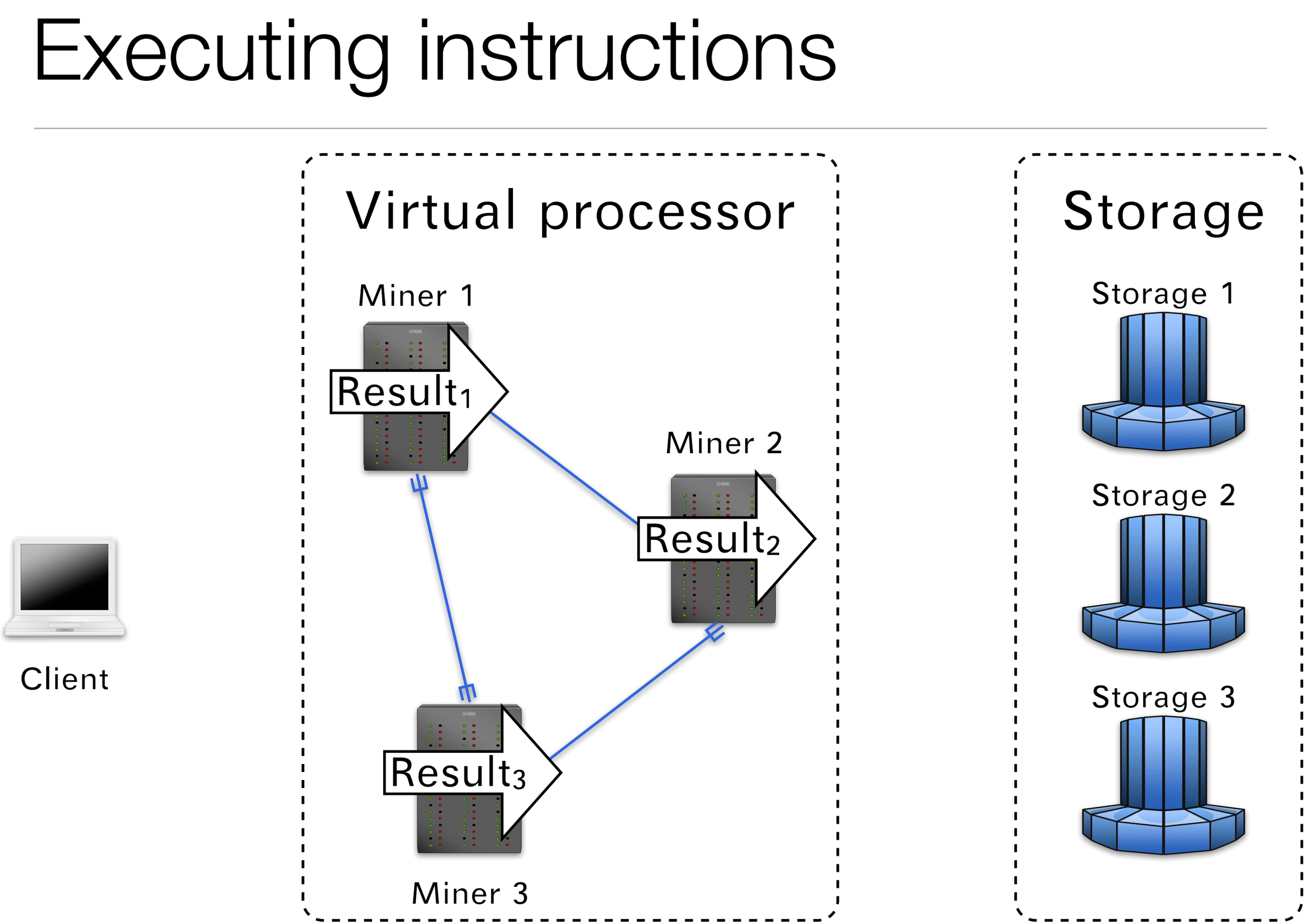
Storage 2



Storage 3



Executing instructions

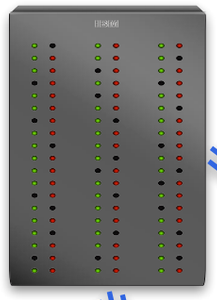


Executing instructions

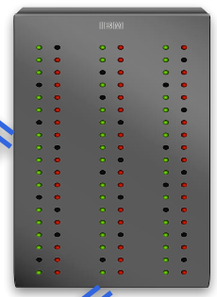
Client

Virtual processor

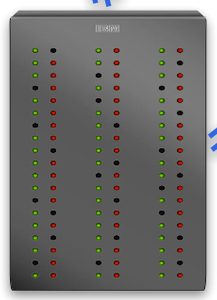
Miner 1



Miner 2



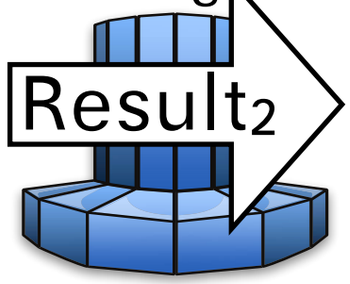
Miner 3



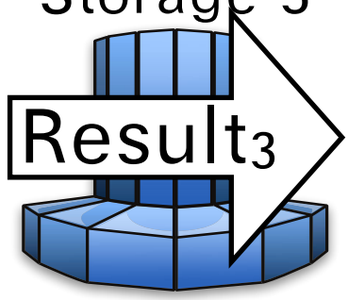
Storage

Storage 1

Storage 2



Storage 3

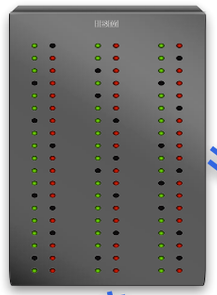


Executing instructions

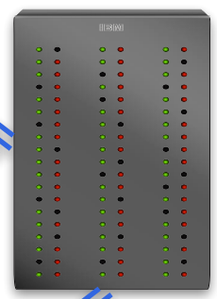
Client

Virtual processor

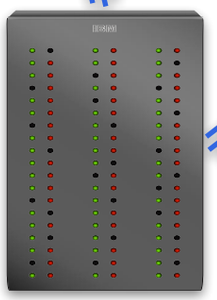
Miner 1



Miner 2

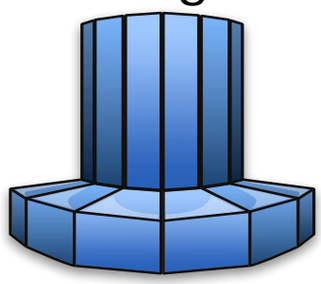


Miner 3

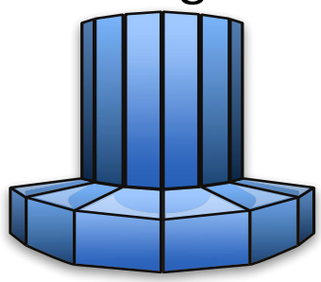


Storage

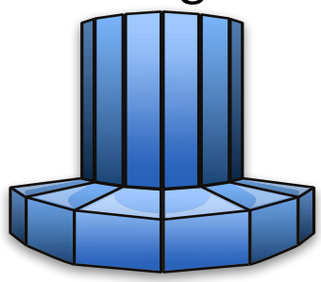
Storage 1



Storage 2



Storage 3



From instructions to code

From instructions to code

- All protocols used to perform operations are universally composable

From instructions to code

- All protocols used to perform operations are universally composable

- That is, we can run them one after the other or in parallel without losing security

From instructions to code

- All protocols used to perform operations are universally composable

- That is, we can run them one after the other or in parallel without losing security

- It follows directly that we can write programs that run on this processor

What needs to be done

n-party protocols for Sharemind

n-party protocols for Sharemind

- Expand Sharemind to support more than three parties.

n-party protocols for Sharemind

- Expand Sharemind to support more than three parties.
- Listed on the webpage, but already taken.

Fix the protocol prover

Fix the protocol prover

- There is a protocol prover written by a NordSecMob student last year. An MSc thesis was written.

Fix the protocol prover

- There is a protocol prover written by a NordSecMob student last year. An MSc thesis was written.

- Improvements are needed. Requires:

Fix the protocol prover

- There is a protocol prover written by a NordSecMob student last year. An MSc thesis was written.

- Improvements are needed. Requires:

  - Java programming knowledge

Fix the protocol prover

- There is a protocol prover written by a NordSecMob student last year. An MSc thesis was written.

- Improvements are needed. Requires:

  - Java programming knowledge

  - an understanding of cryptographic protocols

Fix the protocol prover

- There is a protocol prover written by a NordSecMob student last year. An MSc thesis was written.

- Improvements are needed. Requires:
  - Java programming knowledge
  - an understanding of cryptographic protocols
- Not an MSc topic. Not too tough.

Define the application domain

Define the application domain

- Find out and write down, what kind of applications Sharemind is good for. What kind of security guarantees are achieved in real life?

Define the application domain

- Find out and write down, what kind of applications Sharemind is good for. What kind of security guarantees are achieved in real life?

- Requires:

Define the application domain

- Find out and write down, what kind of applications Sharemind is good for. What kind of security guarantees are achieved in real life?

- Requires:
  - understanding of security and privacy concerns

Define the application domain

- Find out and write down, what kind of applications Sharemind is good for. What kind of security guarantees are achieved in real life?

- Requires:
  - understanding of security and privacy concerns
  - understanding how the world works :)

Define the application domain

- Find out and write down, what kind of applications Sharemind is good for. What kind of security guarantees are achieved in real life?

- Requires:
  - understanding of security and privacy concerns
  - understanding how the world works :)
- Possibly an MSc topic. Less theory, but a lot of work.

Implement data mining algorithms

Implement data mining algorithms

- Sharemind can compute simple stuff. To make it practical we need to implement algorithms.

Implement data mining algorithms

- Sharemind can compute simple stuff. To make it practical we need to implement algorithms.

- Choose, implement and profile some group of algorithms on Sharemind. Requires:

Implement data mining algorithms

- Sharemind can compute simple stuff. To make it practical we need to implement algorithms.

- Choose, implement and profile some group of algorithms on Sharemind. Requires:

  - understanding of Sharemind

Implement data mining algorithms

- Sharemind can compute simple stuff. To make it practical we need to implement algorithms.

- Choose, implement and profile some group of algorithms on Sharemind. Requires:

  - understanding of Sharemind

  - understanding of data mining algorithms

Implement data mining algorithms

- Sharemind can compute simple stuff. To make it practical we need to implement algorithms.

- Choose, implement and profile some group of algorithms on Sharemind. Requires:

  - understanding of Sharemind

  - understanding of data mining algorithms

- A thesis topic, if reaaally needed. Moderately tough.

An information flow analyzer

An information flow analyzer

- There is an assembly language that runs on Sharemind. In the end of 2008, a high level language will appear.

An information flow analyzer

- There is an assembly language that runs on Sharemind. In the end of 2008, a high level language will appear.

- It will contain private and public variables. We need an information flow analyzer that determines whether private information leaks during execution.

An information flow analyzer

- There is an assembly language that runs on Sharemind. In the end of 2008, a high level language will appear.

- It will contain private and public variables. We need an information flow analyzer that determines whether private information leaks during execution.

- Requires understanding of programming languages and privacy. Also requires that you talk to Peeter.

An information flow analyzer

- There is an assembly language that runs on Sharemind. In the end of 2008, a high level language will appear.

- It will contain private and public variables. We need an information flow analyzer that determines whether private information leaks during execution.

- Requires understanding of programming languages and privacy. Also requires that you talk to Peeter.

- Definitely an MSc topic. Moderately tough.

Improve the security model

Improve the security model

- Currently, Sharemind is secure in the honest-but-curious model.

Improve the security model

- Currently, Sharemind is secure in the honest-but-curious model.

- Find out what needs to be done to make the framework secure in the malicious model. Requires:

Improve the security model

- Currently, Sharemind is secure in the honest-but-curious model.

- Find out what needs to be done to make the framework secure in the malicious model. Requires:

  - Advanced knowledge of cryptography and protocols.

Improve the security model

- Currently, Sharemind is secure in the honest-but-curious model.

- Find out what needs to be done to make the framework secure in the malicious model. Requires:

  - Advanced knowledge of cryptography and protocols.

  - C++ programming skills if you want to implement.

Improve the security model

- Currently, Sharemind is secure in the honest-but-curious model.

- Find out what needs to be done to make the framework secure in the malicious model. Requires:

  - Advanced knowledge of cryptography and protocols.

  - C++ programming skills if you want to implement.

- Definitely an MSc topic. Quite hard work.

More info:
http://sharemind.cs.ut.ee/

Contact me.
db@ut.ee