JURIX 2020

Permissioned blockchains: Towards privacy management and data regulation compliance

Authors:
Paulo Henrique Alves
Isabella Z. Frajhof
Fernando A. Correia
Clarisse de Souza
Helio Lopes

PUC RIO
DEPARTAMENTO DE INFORMÁTICA PUC-RIO
ideias
daslab

Research Group

Paulo Henrique Alves
Department of Informatics

Isabella Z. Frajhof
Law School

Fernando A. Correia
Department of Informatics

Clarisse de Souza
Department of Informatics

Helio Lopes
Department of Informatics

Scenario

Influenza

The Pandemic Influenza of 1918 (from: World Health Organization, 2018).

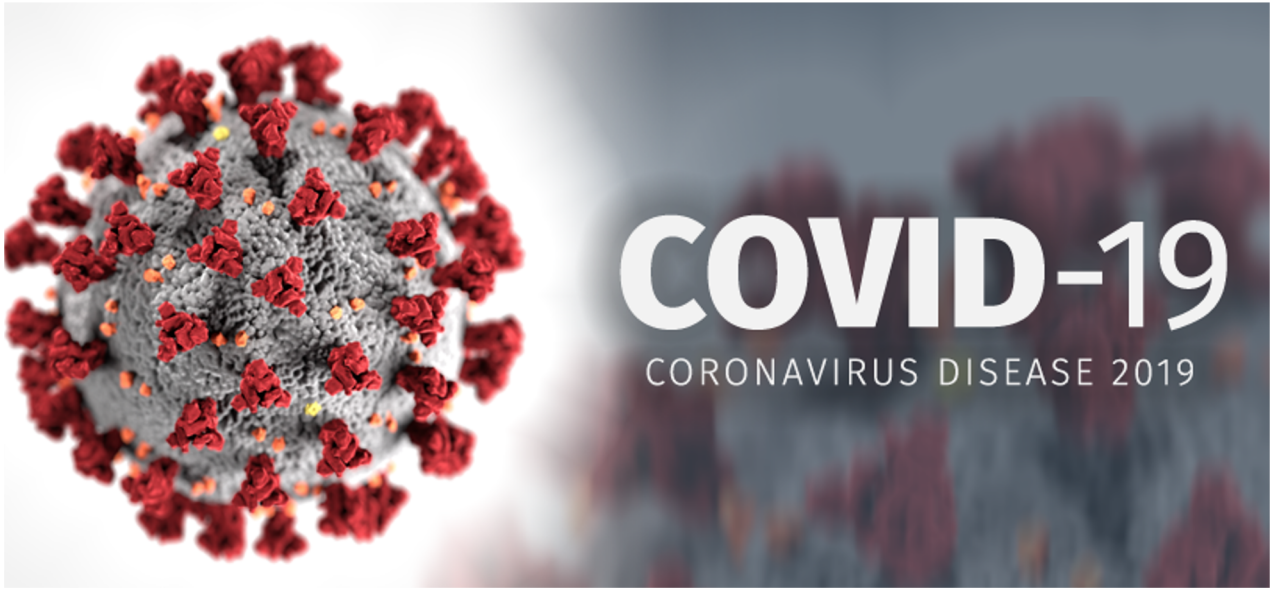
Ebola

Coronavirus and Ebola Together Test (from: The Wall Street Journal, 2020).

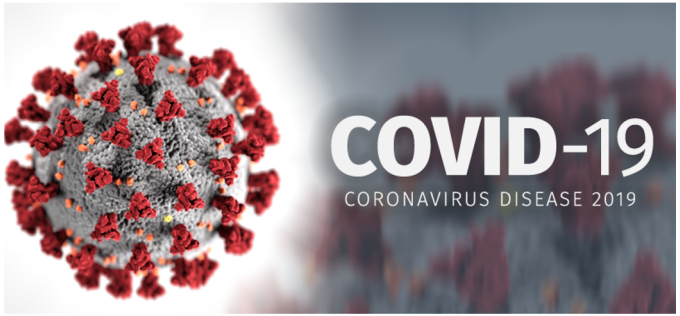
Zika

Zika virus. (from: The Telegraph, 2016).

Scenario



Scenario



Blockchain Technology

	Public Blockchain	Permissioned Blockchain
Access	Anyone	Multiple Organizations
Authority	Decentralized	Decentralized
Consensus	Permissionless	Multi-party Consensus
Data Handling	Read and Write Access for Anyone	Read and Write Access for Multiple Organizations

Health Data Sharing

Brazilian Data Regulation

Purpose Limitation

Data Minimization

Data Anonymization

Transparency

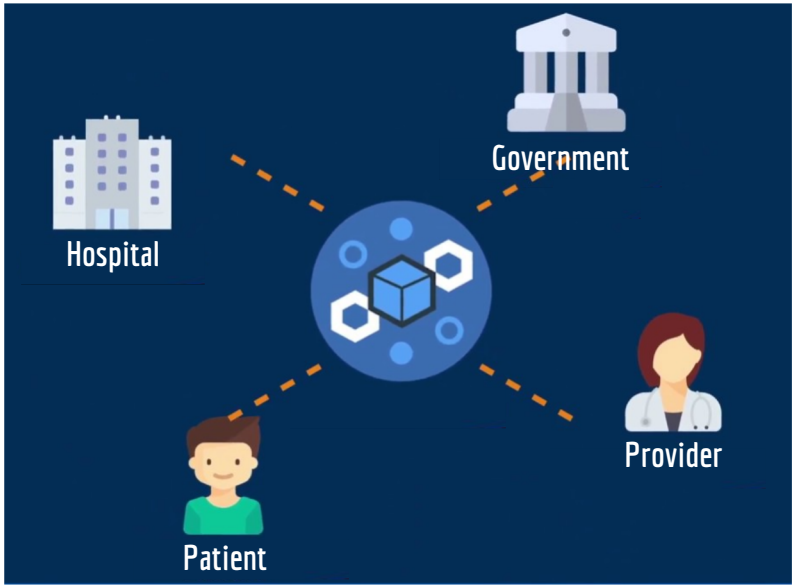
Data Governance

Data Governance is vital to create a well-defined, secure, transparent, and traceable environment to share health.

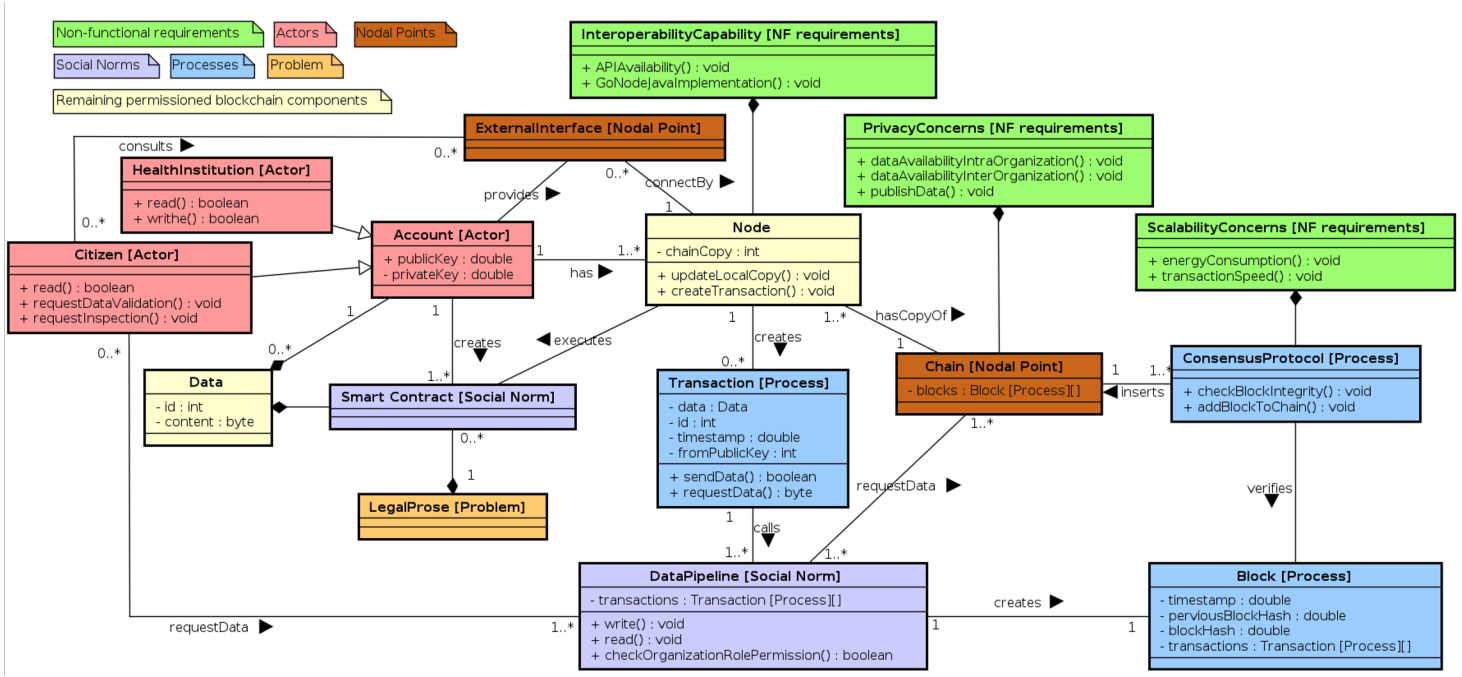
Data Governance

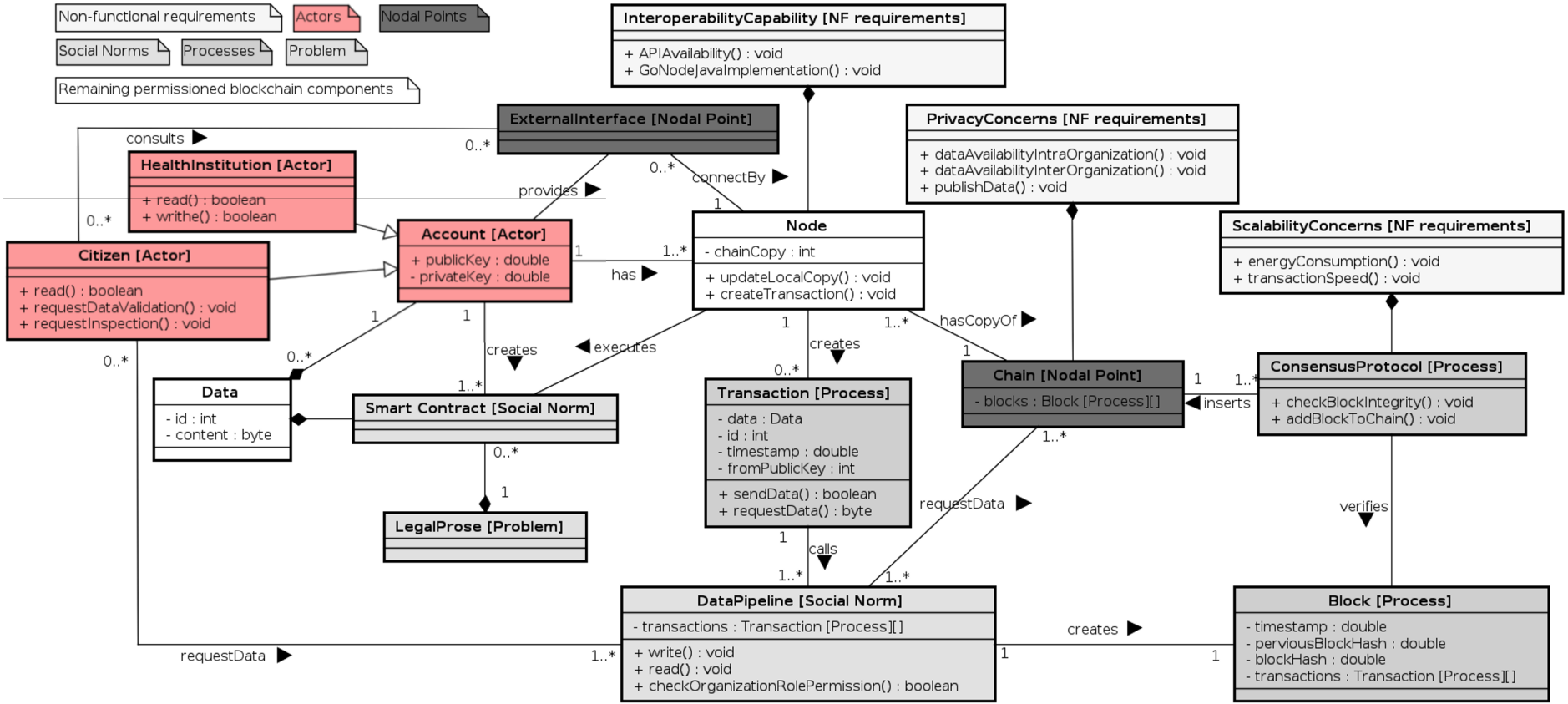
In order to identify the main entities in the pandemic scenario, we based our approach on the Governance Analytical Framework.

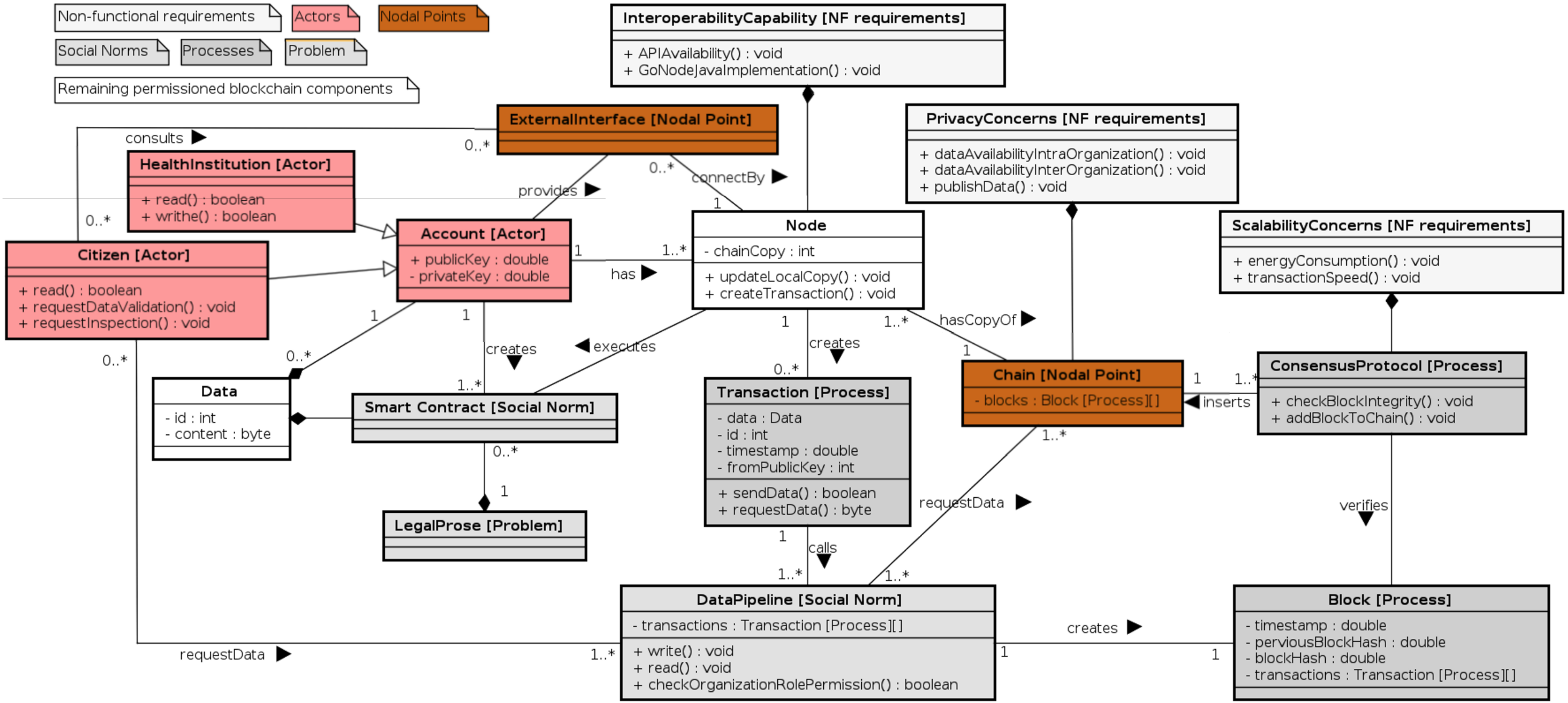
Blockchain and LGPD Data Governance

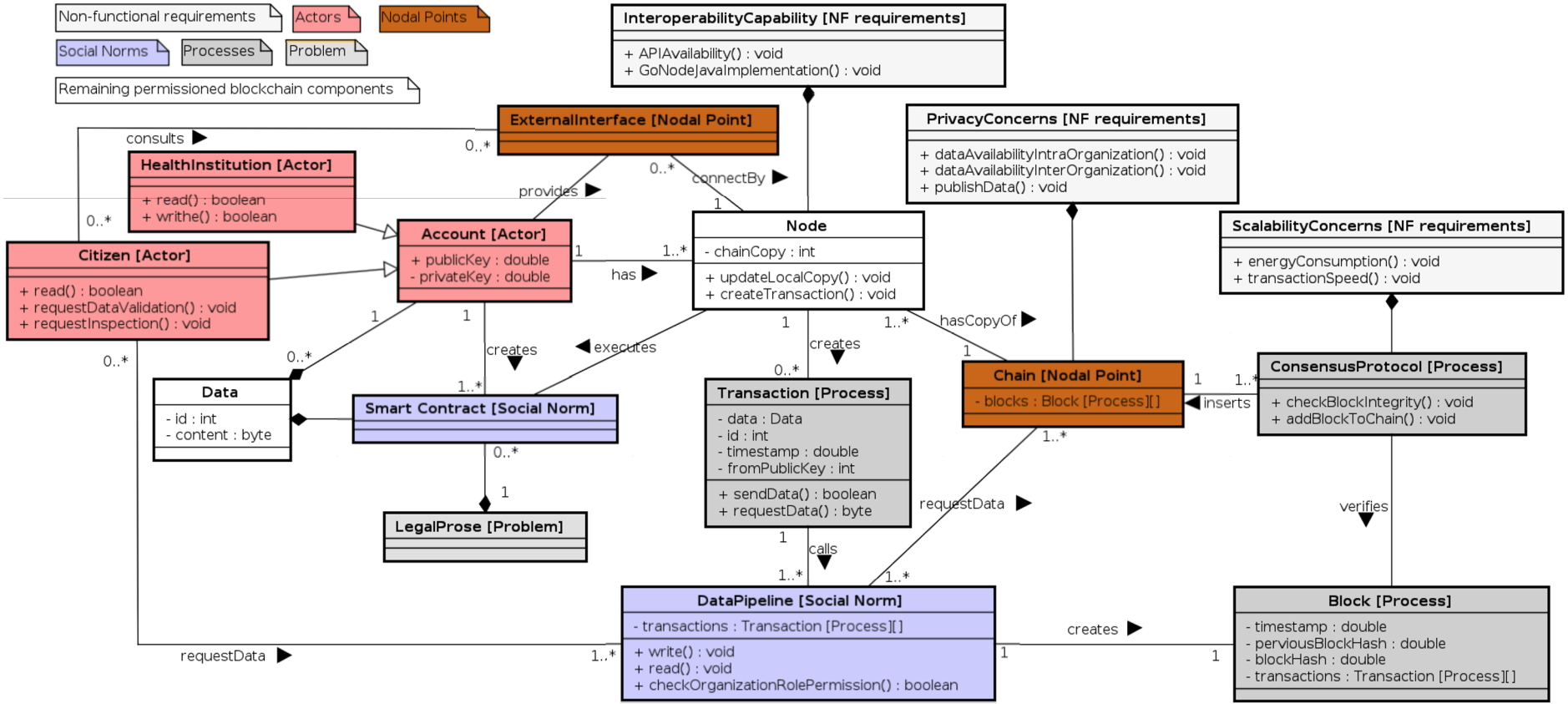


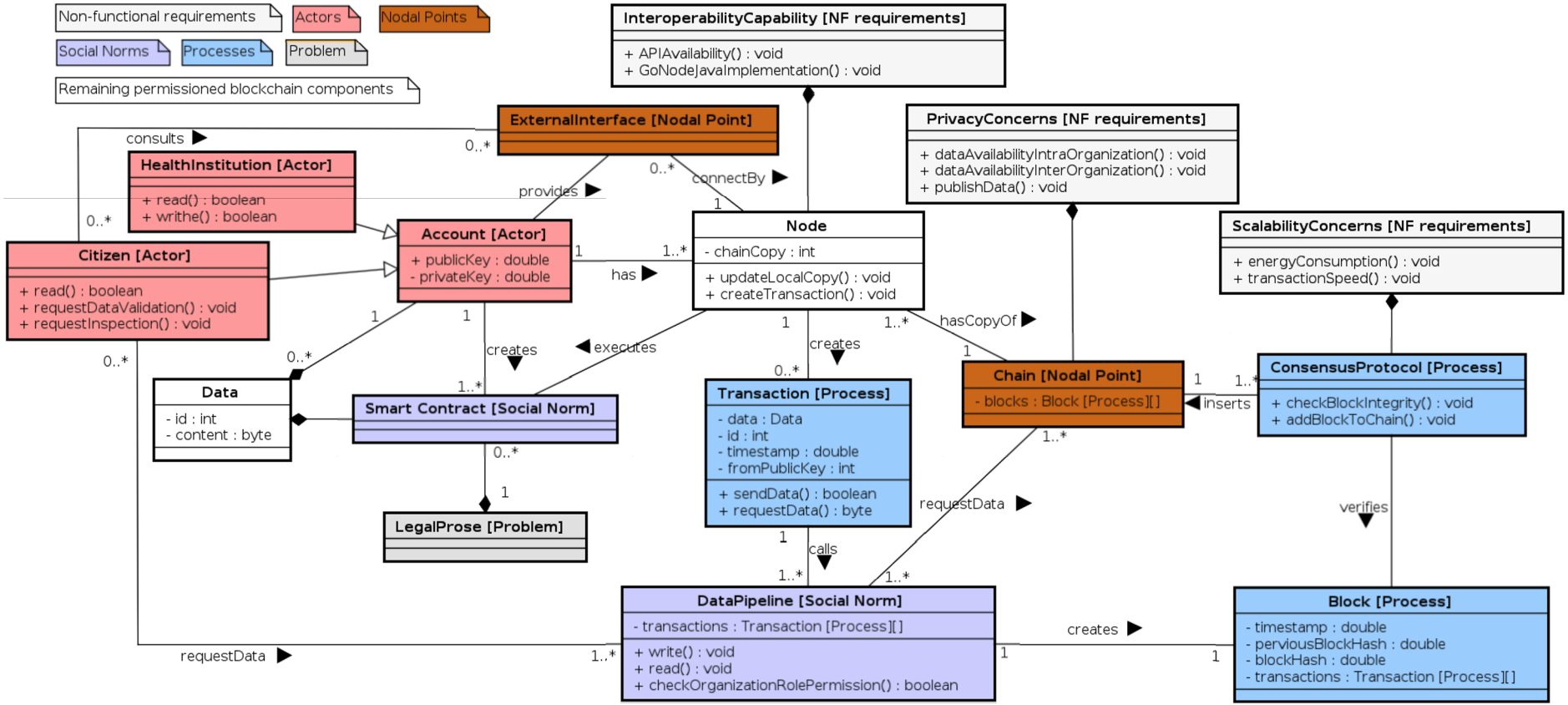
Blockchain and LGPD Data Governance

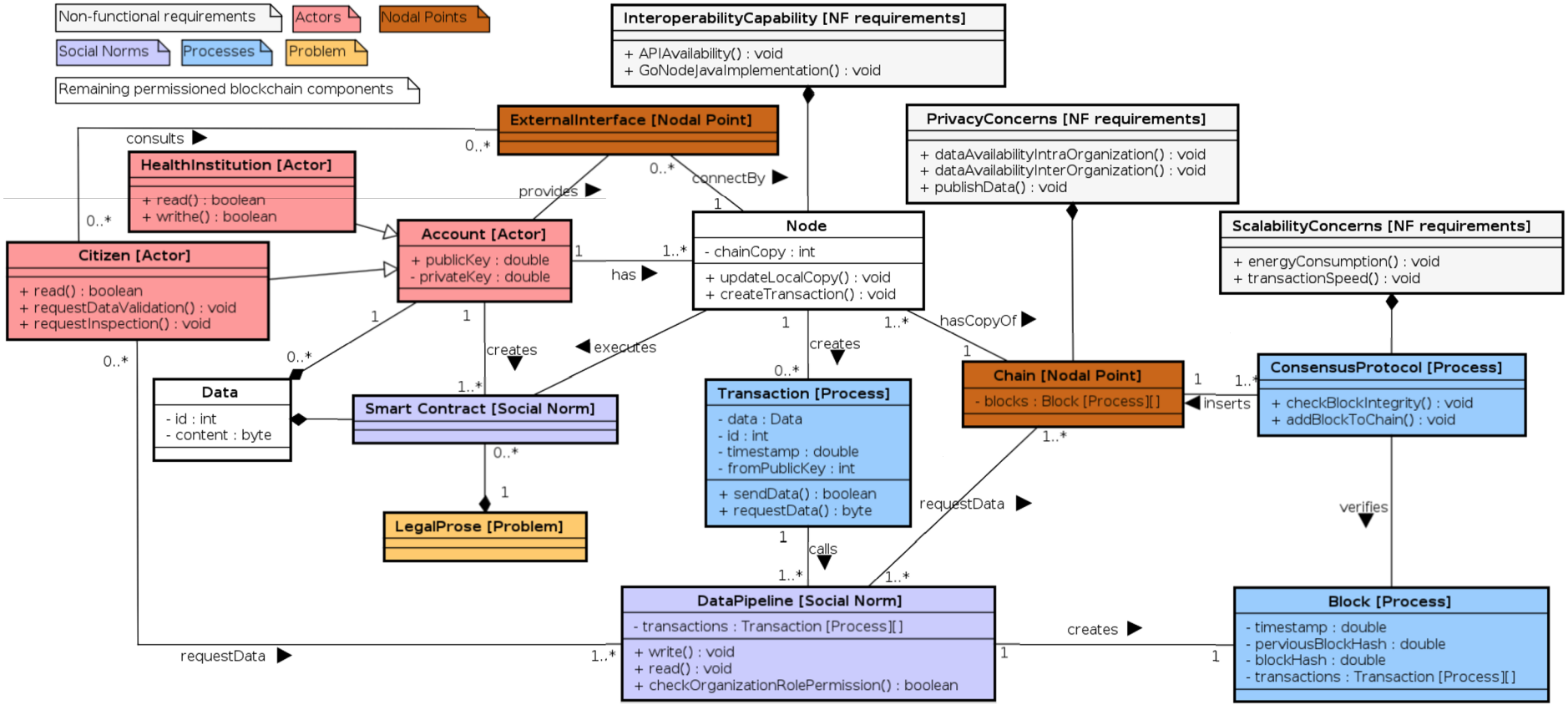


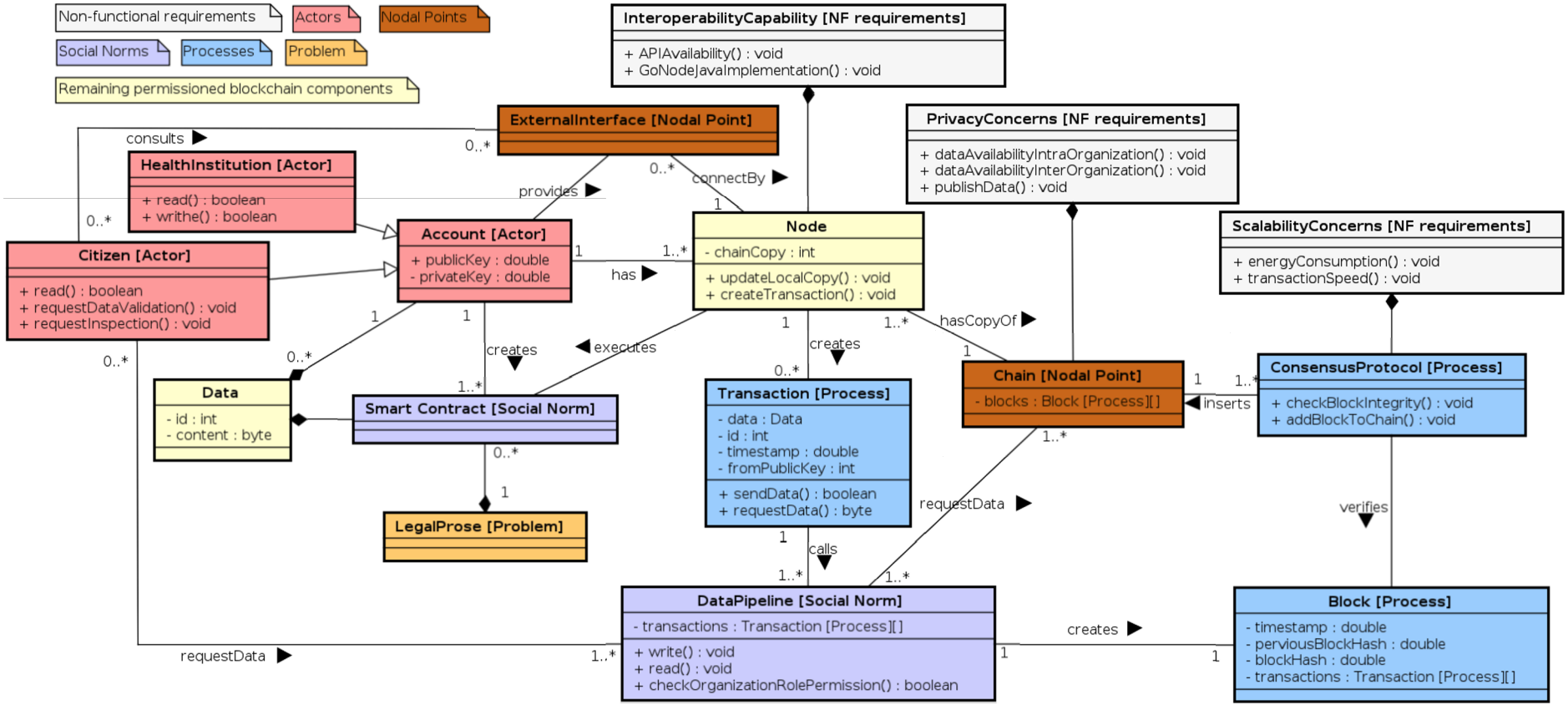

Non-functional requirements
Actors
Nodal Points
Social Norms
Processes
Problem
Remaining permissioned blockchain components
InteroperabilityCapability [NF requirements]
+ APIAvailability() : void
+ GoNodeJavaImplementation() : void
ExternalInterface [Nodal Point]
PrivacyConcerns [NF requirements]
+ dataAvailabilityIntraOrganization() : void
+ dataAvailabilityInterOrganization() : void
+ publishData() : void
consults
0..*
connectBy
provides
HealthInstitution [Actor]
+ read() : boolean
+ writhe() : boolean
Account [Actor]
+ publicKey : double
- privateKey : double
Node
- chainCopy : int
+ updateLocalCopy() : void
+ createTransaction() : void
ScalabilityConcerns [NF requirements]
+ energyConsumption() : void
+ transactionSpeed() : void
Citizen [Actor]
+ read() : boolean
+ requestDataValidation() : void
+ requestInspection() : void
has
1
1..*
creates
executes
hasCopyOf
Data
- id : int
- content : byte
Smart Contract [Social Norm]
Transaction [Process]
- data : Data
- id : int
- timestamp : double
- fromPublicKey : int
+ sendData() : boolean
+ requestData() : byte
Chain [Nodal Point]
- blocks : Block [Process][]
inserts
ConsensusProtocol [Process]
+ checkBlockIntegrity() : void
+ addBlockToChain() : void
LegalProse [Problem]
requestData
calls
verifies
DataPipeline [Social Norm]
- transactions : Transaction [Process][]
+ write() : void
+ read() : void
+ checkOrganizationRolePermission() : boolean
creates
Block [Process]
- timestamp : double
- perviousBlockHash : double
- blockHash : double
- transactions : Transaction [Process][]

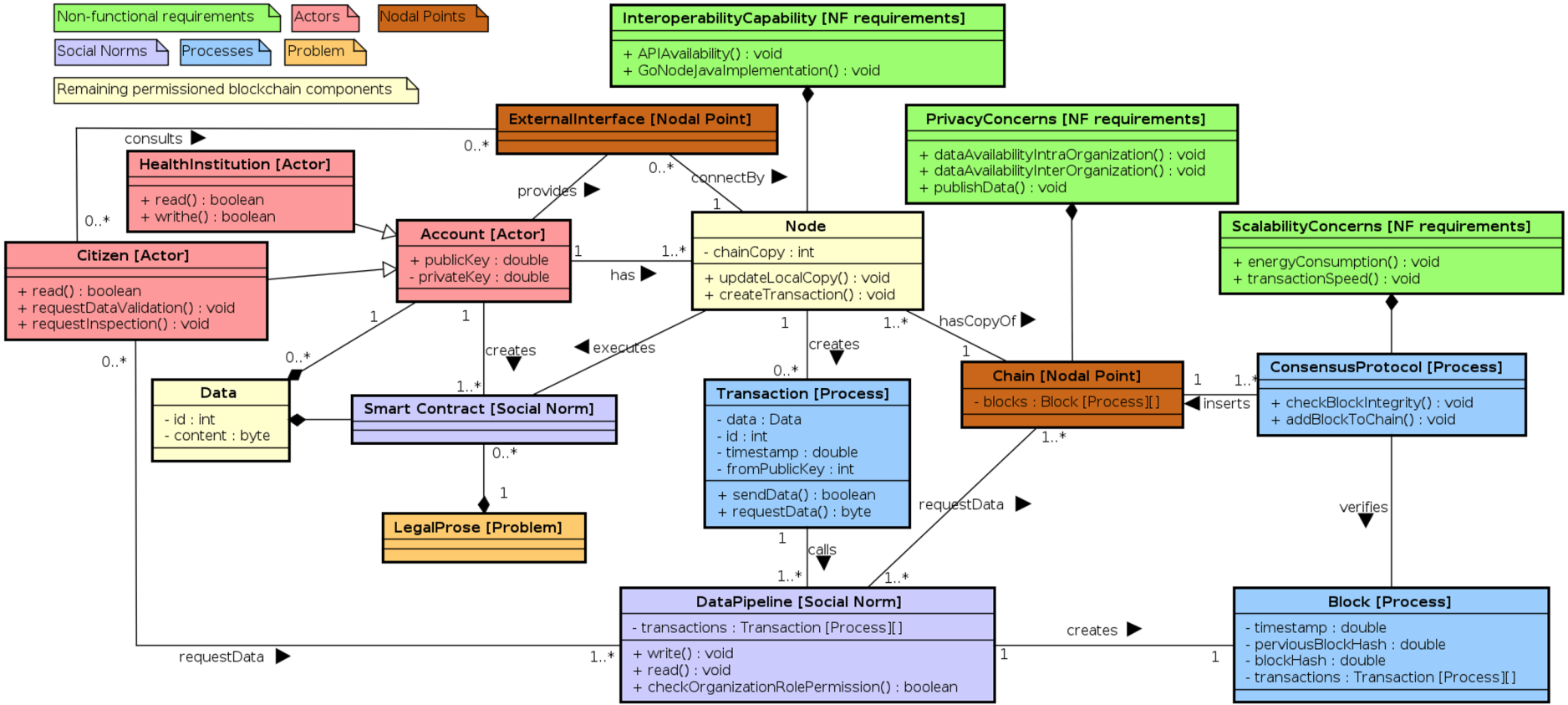

Non-functional requirements
Actors
Nodal Points
Social Norms
Processes
Problem
Remaining permissioned blockchain components
InteroperabilityCapability [NF requirements]
+ APIAvailability() : void
+ GoNodeJavaImplementation() : void
ExternalInterface [Nodal Point]
PrivacyConcerns [NF requirements]
+ dataAvailabilityIntraOrganization() : void
+ dataAvailabilityInterOrganization() : void
+ publishData() : void
ScalabilityConcerns [NF requirements]
+ energyConsumption() : void
+ transactionSpeed() : void
consults
HealthInstitution [Actor]
+ read() : boolean
+ writhe() : boolean
Citizen [Actor]
+ read() : boolean
+ requestDataValidation() : void
+ requestInspection() : void
Account [Actor]
+ publicKey : double
- privateKey : double
provides
connectBy
has
Node
- chainCopy : int
+ updateLocalCopy() : void
+ createTransaction() : void
hasCopyOf
executes
creates
Data
- id : int
- content : byte
Smart Contract [Social Norm]
LegalProse [Problem]
Transaction [Process]
- data : Data
- id : int
- timestamp : double
- fromPublicKey : int
+ sendData() : boolean
+ requestData() : byte
Chain [Nodal Point]
- blocks : Block [Process][]
inserts
ConsensusProtocol [Process]
+ checkBlockIntegrity() : void
+ addBlockToChain() : void
requestData
verifies
calls
DataPipeline [Social Norm]
- transactions : Transaction [Process][]
+ write() : void
+ read() : void
+ checkOrganizationRolePermission() : boolean
Block [Process]
- timestamp : double
- perviousBlockHash : double
- blockHash : double
- transactions : Transaction [Process][]
0..*
1
1..*


Non-functional requirements
Actors
Nodal Points
Social Norms
Processes
Problem
Remaining permissioned blockchain components
InteroperabilityCapability [NF requirements]
+ APIAvailability() : void
+ GoNodeJavaImplementation() : void
ExternalInterface [Nodal Point]
PrivacyConcerns [NF requirements]
+ dataAvailabilityIntraOrganization() : void
+ dataAvailabilityInterOrganization() : void
+ publishData() : void
ScalabilityConcerns [NF requirements]
+ energyConsumption() : void
+ transactionSpeed() : void
consults
HealthInstitution [Actor]
+ read() : boolean
+ writhe() : boolean
provides
connectBy
Account [Actor]
+ publicKey : double
- privateKey : double
Node
- chainCopy : int
+ updateLocalCopy() : void
+ createTransaction() : void
has
Citizen [Actor]
+ read() : boolean
+ requestDataValidation() : void
+ requestInspection() : void
creates
executes
hasCopyOf
Data
- id : int
- content : byte
Smart Contract [Social Norm]
Transaction [Process]
- data : Data
- id : int
- timestamp : double
- fromPublicKey : int
+ sendData() : boolean
+ requestData() : byte
Chain [Nodal Point]
- blocks : Block [Process][]
inserts
ConsensusProtocol [Process]
+ checkBlockIntegrity() : void
+ addBlockToChain() : void
LegalProse [Problem]
requestData
calls
verifies
DataPipeline [Social Norm]
- transactions : Transaction [Process][]
+ write() : void
+ read() : void
+ checkOrganizationRolePermission() : boolean
creates
Block [Process]
- timestamp : double
- perviousBlockHash : double
- blockHash : double
- transactions : Transaction [Process][]


Non-functional requirements
Actors
Nodal Points
Social Norms
Processes
Problem
Remaining permissioned blockchain components
InteroperabilityCapability [NF requirements]
+ APIAvailability() : void
+ GoNodeJavaImplementation() : void
ExternalInterface [Nodal Point]
PrivacyConcerns [NF requirements]
+ dataAvailabilityIntraOrganization() : void
+ dataAvailabilityInterOrganization() : void
+ publishData() : void
ScalabilityConcerns [NF requirements]
+ energyConsumption() : void
+ transactionSpeed() : void
consults
HealthInstitution [Actor]
+ read() : boolean
+ writhe() : boolean
provides
connectBy
Account [Actor]
+ publicKey : double
- privateKey : double
Node
- chainCopy : int
+ updateLocalCopy() : void
+ createTransaction() : void
has
Citizen [Actor]
+ read() : boolean
+ requestDataValidation() : void
+ requestInspection() : void
creates
executes
hasCopyOf
Data
- id : int
- content : byte
Smart Contract [Social Norm]
Transaction [Process]
- data : Data
- id : int
- timestamp : double
- fromPublicKey : int
+ sendData() : boolean
+ requestData() : byte
Chain [Nodal Point]
- blocks : Block [Process][]
inserts
ConsensusProtocol [Process]
+ checkBlockIntegrity() : void
+ addBlockToChain() : void
LegalProse [Problem]
requestData
calls
verifies
DataPipeline [Social Norm]
- transactions : Transaction [Process][]
+ write() : void
+ read() : void
+ checkOrganizationRolePermission() : boolean
creates
Block [Process]
- timestamp : double
- perviousBlockHash : double
- blockHash : double
- transactions : Transaction [Process][]


Non-functional requirements
Actors
Nodal Points
Social Norms
Processes
Problem
Remaining permissioned blockchain components
InteroperabilityCapability [NF requirements]
+ APIAvailability() : void
+ GoNodeJavaImplementation() : void
ExternalInterface [Nodal Point]
PrivacyConcerns [NF requirements]
+ dataAvailabilityIntraOrganization() : void
+ dataAvailabilityInterOrganization() : void
+ publishData() : void
ScalabilityConcerns [NF requirements]
+ energyConsumption() : void
+ transactionSpeed() : void
consults
HealthInstitution [Actor]
+ read() : boolean
+ writhe() : boolean
provides
connectBy
Account [Actor]
+ publicKey : double
- privateKey : double
Node
- chainCopy : int
+ updateLocalCopy() : void
+ createTransaction() : void
has
Citizen [Actor]
+ read() : boolean
+ requestDataValidation() : void
+ requestInspection() : void
creates
executes
hasCopyOf
Data
- id : int
- content : byte
Smart Contract [Social Norm]
Transaction [Process]
- data : Data
- id : int
- timestamp : double
- fromPublicKey : int
+ sendData() : boolean
+ requestData() : byte
Chain [Nodal Point]
- blocks : Block [Process][]
inserts
ConsensusProtocol [Process]
+ checkBlockIntegrity() : void
+ addBlockToChain() : void
LegalProse [Problem]
requestData
calls
verifies
DataPipeline [Social Norm]
- transactions : Transaction [Process][]
+ write() : void
+ read() : void
+ checkOrganizationRolePermission() : boolean
creates
Block [Process]
- timestamp : double
- perviousBlockHash : double
- blockHash : double
- transactions : Transaction [Process][]


Non-functional requirements
Actors
Nodal Points
Social Norms
Processes
Problem
Remaining permissioned blockchain components
InteroperabilityCapability [NF requirements]
+ APIAvailability() : void
+ GoNodeJavaImplementation() : void
ExternalInterface [Nodal Point]
PrivacyConcerns [NF requirements]
+ dataAvailabilityIntraOrganization() : void
+ dataAvailabilityInterOrganization() : void
+ publishData() : void
ScalabilityConcerns [NF requirements]
+ energyConsumption() : void
+ transactionSpeed() : void
consults
HealthInstitution [Actor]
+ read() : boolean
+ writhe() : boolean
provides
connectBy
Account [Actor]
+ publicKey : double
- privateKey : double
Node
- chainCopy : int
+ updateLocalCopy() : void
+ createTransaction() : void
has
Citizen [Actor]
+ read() : boolean
+ requestDataValidation() : void
+ requestInspection() : void
creates
executes
hasCopyOf
Data
- id : int
- content : byte
Smart Contract [Social Norm]
Transaction [Process]
- data : Data
- id : int
- timestamp : double
- fromPublicKey : int
+ sendData() : boolean
+ requestData() : byte
Chain [Nodal Point]
- blocks : Block [Process][]
inserts
ConsensusProtocol [Process]
+ checkBlockIntegrity() : void
+ addBlockToChain() : void
LegalProse [Problem]
requestData
calls
verifies
DataPipeline [Social Norm]
- transactions : Transaction [Process][]
+ write() : void
+ read() : void
+ checkOrganizationRolePermission() : boolean
creates
Block [Process]
- timestamp : double
- perviousBlockHash : double
- blockHash : double
- transactions : Transaction [Process][]


Non-functional requirements
Actors
Nodal Points
Social Norms
Processes
Problem
Remaining permissioned blockchain components
InteroperabilityCapability [NF requirements]
+ APIAvailability() : void
+ GoNodeJavaImplementation() : void
ExternalInterface [Nodal Point]
PrivacyConcerns [NF requirements]
+ dataAvailabilityIntraOrganization() : void
+ dataAvailabilityInterOrganization() : void
+ publishData() : void
ScalabilityConcerns [NF requirements]
+ energyConsumption() : void
+ transactionSpeed() : void
consults
0..*
HealthInstitution [Actor]
+ read() : boolean
+ writhe() : boolean
provides
connectBy
Citizen [Actor]
+ read() : boolean
+ requestDataValidation() : void
+ requestInspection() : void
Account [Actor]
+ publicKey : double
- privateKey : double
has
Node
- chainCopy : int
+ updateLocalCopy() : void
+ createTransaction() : void
hasCopyOf
creates
executes
Data
- id : int
- content : byte
Smart Contract [Social Norm]
Transaction [Process]
- data : Data
- id : int
- timestamp : double
- fromPublicKey : int
+ sendData() : boolean
+ requestData() : byte
Chain [Nodal Point]
- blocks : Block [Process][]
inserts
ConsensusProtocol [Process]
+ checkBlockIntegrity() : void
+ addBlockToChain() : void
LegalProse [Problem]
requestData
calls
verifies
DataPipeline [Social Norm]
- transactions : Transaction [Process][]
+ write() : void
+ read() : void
+ checkOrganizationRolePermission() : boolean
Block [Process]
- timestamp : double
- perviousBlockHash : double
- blockHash : double
- transactions : Transaction [Process][]

Conclusion

LGPD based blockchain architecture.

Blockchain is a promising technology for data sharing.

COVID-19 application.

First step towards technology model for data regulation compliance.

Further approaches would use this architecture in other health scenarios.

Thank you

Paulo Henrique Alves - palves@inf.puc-rio.br
PUC-Rio, Brazil, 2020