**Crypto Munt**

**Software Architectuur Document**

**Versie 1.0**

**Documenthistorie**

| **Datum** | **Versie** | **Beschrijving** | **Auteur** |
| --- | --- | --- | --- |
| 14/10/2021 | 0.1 | Initiële versie | Ivo Gerner |
| 30/10/2021 | 1.0 | 1e oplevering | Ivo Gerner |
|  |  |  |  |
|  |  |  |  |

**Inhoudsopgave**

[**Inleiding**](#_heading=h.9bbsmjycf9ce) **3**

[Doel van dit document](#_heading=h.1fob9te) 4

[Referenties](#_heading=h.h64byd3tersk) 4

[Documentoverzicht](#_heading=h.filnido1en3y) 4

[**Architecturele eisen**](#_heading=h.tyjcwt) **4**

[Niet-functionele requirements](#_heading=h.g3zawsxcj8gb) 5

[Use Case View (functionele requirements)](#_heading=h.qxvnd8re03th) 5

[**Logical View**](#_heading=h.ywbjxy81iy05) **5**

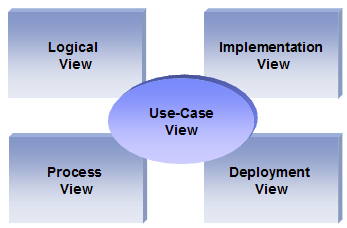
[Lagen](#_heading=h.1zhsw6h91ucq) 6

# **Inleiding**

## **Doel van dit document**

Het Software Architectuur Document (SAD) bevat een uitgebreide architecturele kijk op het systeem Crypto Muntontwikkeld door Jacob de Boer. Het beschrijft een aantal verschillende architecturele views van het systeem om zo verschillende aspecten van het systeem te belichten.

Dit document beschrijft de verschillende RUP-views op de software architectuur volgens het 4+1 view model.



Het 4+1 view model stelt de verschillende belanghebbenden in staat vanuit hun eigen perspectief de invloed van de gekozen architectuur te bepalen. De Process View (communicatie van processen) is niet als los hoofdstuk uitgewerkt maar ondergebracht bij de hoofdstukken 3.3 en 5.

## **Referenties**

| **Titel** | **Versie** | **Auteur** | **Vindplaats** |
| --- | --- | --- | --- |
| Acceptatieplan | 1.2 | Batul Khatib |  |
| Use Case Model | 1.0 | Mike van der Velde |  |
|  |  |  |  |

## **Documentoverzicht**

| **Hoofdstuk** | **Belanghebbende** | **Doel** |
| --- | --- | --- |
| 2 Architecturele eisen | Software Architect | Overzicht van architectureel relevante requirements. |
| 3 Logical View | Programmeur (t.b.v. technisch ontwerp) | Inzicht in de conceptuele structuur van de applicatie. |
| 4 Implementation View | Programmeur (t.b.v. bouw) | Inzicht in de technische structuur van de applicatie. |
| 5 Deployment View | Beheer rollen | Inzicht in de manier waarop de applicatie wordt gedeployed en de manier waarop de (interne en externe) communicatie plaatsvindt. |

# **Architecturele eisen**

## **Niet-functionele requirements**

| **Bron** | **Naam** | **Architecturele relevantie** | **Geadresseerd in:** |
| --- | --- | --- | --- |
| Jacob de Boer | Anonimiteit | In plaats van hashes te genereren voor de id van een gebruiker om deze te refereren, kan deze ook een naam toegewezen krijgen die makkelijker leesbaar is. |  |
| Jacob de Boer | Publieke adressen | Grote complexe adressen zijn voor deze Proof of Concept niet nodig, een 32 bit adres is voldoende |  |
| Jacob de Boer | JSON formaat | Door alle onderlinge berichten van de nodes in JSON weer te geven kunnen derden applicaties maken |  |

## **Use Case View (functionele requirements)**

Onderstaand overzicht verwijst naar architectureel relevante Use Cases uit het Use Case Model (zie referenties).

| **Bron** | **Naam** | **Architecturele relevantie** | **Geadresseerd in:** |
| --- | --- | --- | --- |
|  | UC01: Saldo bekijken in een portemonee | Hierbij moet een call gemaakt worden naar het huidige bevestigde saldo en er moet een GUI zijn om dit in te zien |  |
|  | UC03: Transacties | Er moet een verbinding zijn tussen de betrokken nodes om transacties waar te maken |  |
|  | UC04: Betalingsverzoek middels QR-code | Er moet ondersteuning zijn voor betaalverzoeken, waarbij ook hier een verbinding aanwezig moet zijn tussen de betrokken nodes om transacties waar te maken. |  |
|  | UC05: Transacties inzien | Er moet kunnen worden nagegaan in hoeverre de getoonde transacties geldig zijn. |  |
|  | UC06:Saldo bekijken | Er moet kunnen worden nagegaan in hoeverre de getoonde transacties geldig zijn. |  |

# **Logical View**

De logical view van de CryptoMunt bestaat uit de volgende 4 lagen: User Interface, Network Services, Transaction Services en Ledger Objects

De User Interface beschikt over methoden waarmee de actors communiceren met het systeem.

De Network Services handelt de communicatie tussen de nodes af; ontdekken nodes bij opstarten netwerk, transacties bijhouden en de gehele P2P verbinding.

De Transaction Services valideren transacties d.m.v. consensus algoritme, garandeert authenticiteit en integriteit en kondigt goedgekeurde transacties aan.

Het Ledger Object is de blockchain, waarin alle transacties worden opgeslagen.

## **Lagen**

* *Presentatie:* Dit wordt bewerkstelligd met de User Interface; deze communiceert indirect met de Network Services
* *Service:* Dit zijn de Network Services en de Transaction Services. De Transaction Services communiceert met de Network Services en de Network Services communiceert indirect met de User Interface en met het Ledger Object
* *Domein:* REST-API
* *Data:* Alle transacties worden bijgehouden in het Ledger Object; communicatie vindt plaats met de Network Servies