

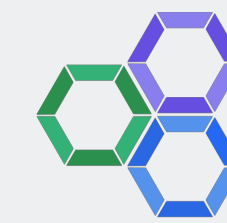
# HTK PILOT PROJEKT

---

**KL**

**KOMBIT**

 **Høje-Taastrup  
Kommune**

 **NORDIC  
BLOCKCHAIN  
ASSOCIATION**

# VELKOMMEN TIL

---

## **1. Indledning til projektet (10 minutter)**

## **2. Gennemgang af Høje Taastrup kommune case**

- Oprettelse af en smart kontrakt (10 min)
- Forløbet når den er aktiv (10 min)
- Ophøring af en smart kontrakt (10 min)

## **3. Blockchain Q&A (≈30 minutter)**

# INDLEDNING

# INDLEDNING

---

## Nordic Blockchain Association

- Non-profit
- Promovering af blockchain teknologien
- over 40 events afholdt
- Trends (fra vores perspektiv)

# INDLEDNING

---

## Hvad er pilot-projekt initiativet?

- Indledende fase af blockchain
- Inspirerer samarbejdspartnere
- Håndgribeliggører teknologien
- Udbygger et framework for aktøren

# INDLEDNING

---

## Hvorfor lave et pilot-projekt i stedet for det fulde produkt?

- Beviser en udfordring kan klares på en ny og bedre måde
- Teknologien er stadig spæd
- Reguleringer er endnu ikke klarlagt
- Kontrakt med KL & KOMBIT

# AFGRÆNSNING

---

## Hvad vi leverer (ifølge aftalen)

- En bedre forståelse for blockchain teknologien og hvordan den kan udnyttes i den offentlige sektor
- En funktionel prototype, der kan bruges som afsæt til at bygge den fulde løsning i fremtiden.
- En rapport som belyser pilot projektets proces, samt muligheder og udfordringer ved teknologien





# AFGRÆNSNING

---

## Hvad vi leverer (ifølge aftalen)

- En bedre forståelse for blockchain teknologien og hvordan den kan udnyttes i den offentlige sektor
- En funktionel prototype, der kan bruges som afsæt til at bygge den fulde løsning i fremtiden.
- En rapport som belyser pilot projektets proces, samt muligheder og udfordringer ved teknologien

# AFGRÆNSNING

---

## Hvad vi leverer udover aftalen

- Visuelle flows og modeller (i rapporten)
- Interaktiv prototype af *potentiel* brugergrænseflade

# AFGRÆNSNING

---

## Hvad vi ikke leverer

- Feasibility analyse
- Implementering

# AFGRÆNSNING

---



# AFGRÆNSNING

---

Feasibility ← Proof of Concept ← Prototype

# CASE

# PROBLEMSTILLING

---

## Identificerede udfordringer

- Høj kompleksitet og store variationer i aftaler
- Mangel på “serve-ret” ved indgåelse af aftaler
- Ressourcekrævende at validere kontrakter

# PRÆMISSER

---

## Vores løsning antager at...

- Kommunernes blockchain indeholder penge/kredit (ikke bitcoin, men kroner)
- Alle aktører har et "blockchainID" (ligesom CVR nummer)
- Smart kontrakt = aftale + kontrakt + transaktioner



# LØSNINGEN

Beskrevet i tre dele...

**Del 1**

Kontrakten oprettes

**Del 2**

Kontrakten er aktiv

**Del 3**

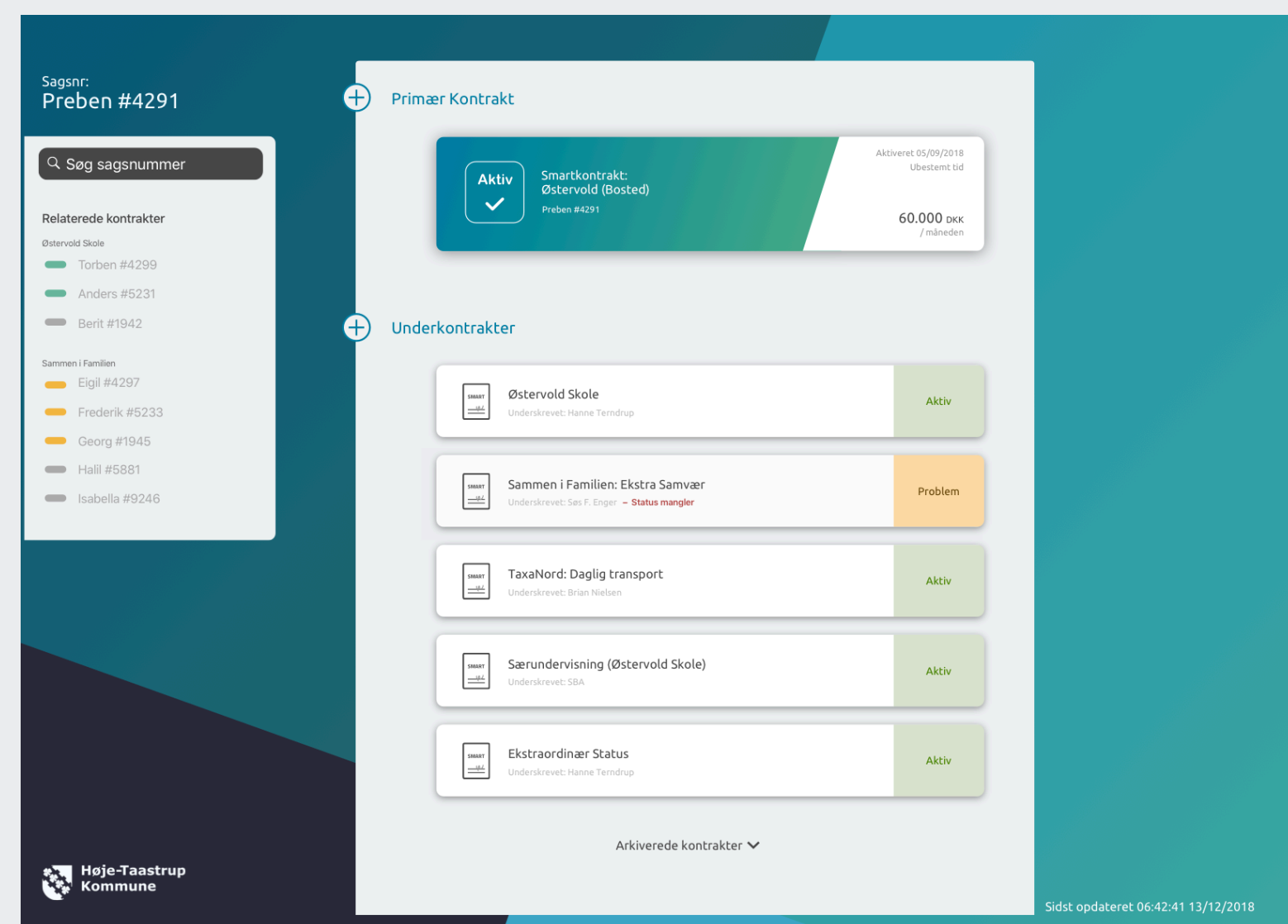
Kontrakten ophører

# LØSNINGEN

...og i tre niveauer



Illustreret



Brugergrænseflade

```
/**
 * Mine pending transactions and create new transaction for mining reward.
 * @param {string} rewardAddress
 */
const mine = async (req, res) => {
  codeCoin.minePendingTransactions(req.body.rewardAddress);

  // Notify other blockchains a new block is added
  let promiseArray = [];

  codeCoin.nodes.map(node => {
    let promise = rp.get({
      uri: `http://localhost:${node}/events/blockchain/update`,
      json: true
    });

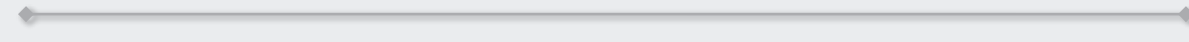
    promiseArray.push(promise);
  })

  await Promise.all(promiseArray);

  res.send("Mining finished. Reward Transaction created.");
};
```

Kode

# DEL 1: KONTRAKTEN OPRETTES



**Koncept illustreret**

# DEL 1: KONTRAKTEN OPRETTES

---

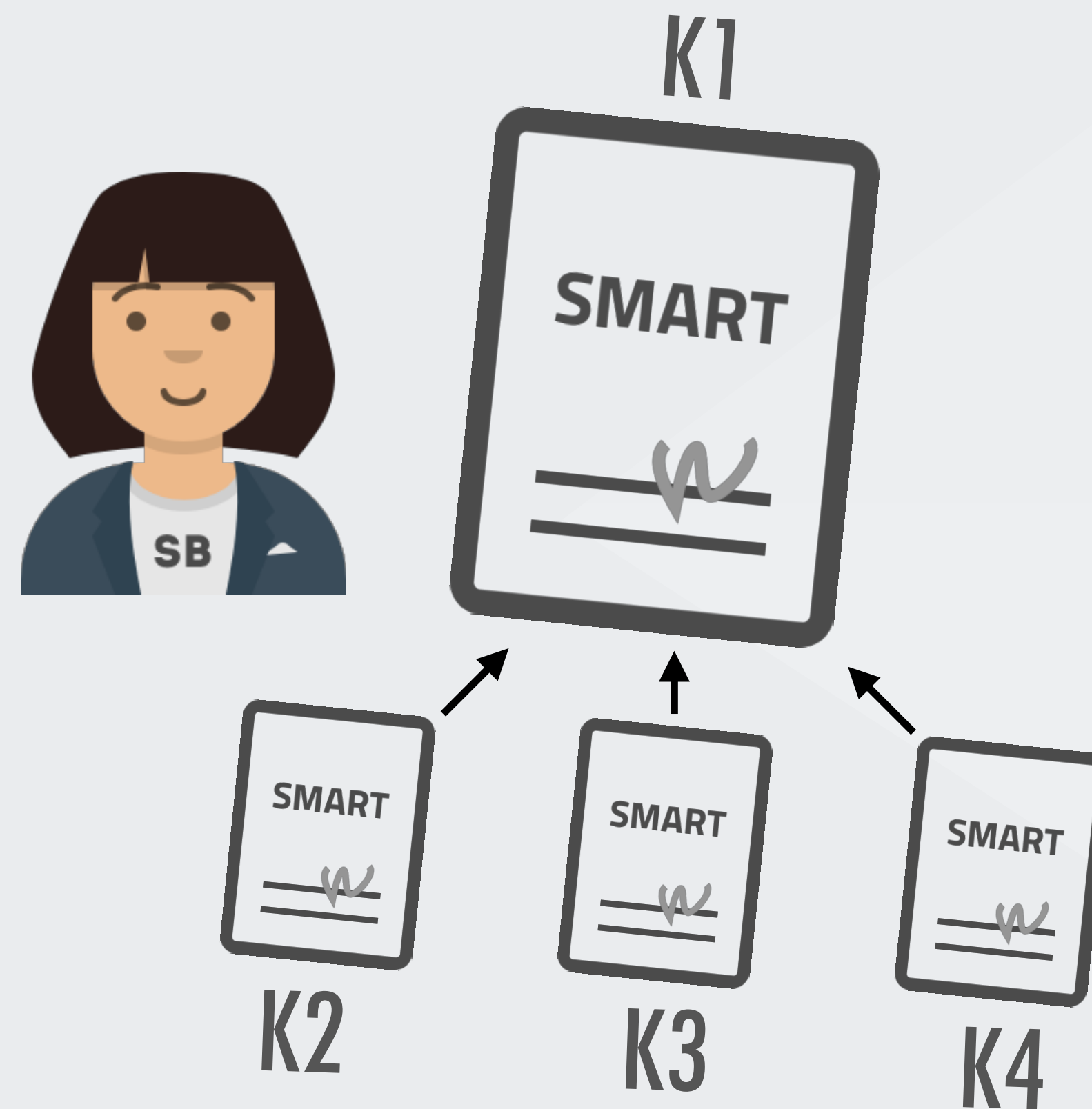
Det besluttet at Preben skal  
anbringes uden for hjemmet.



Skabelon:  
Hvem har plads til Preben?

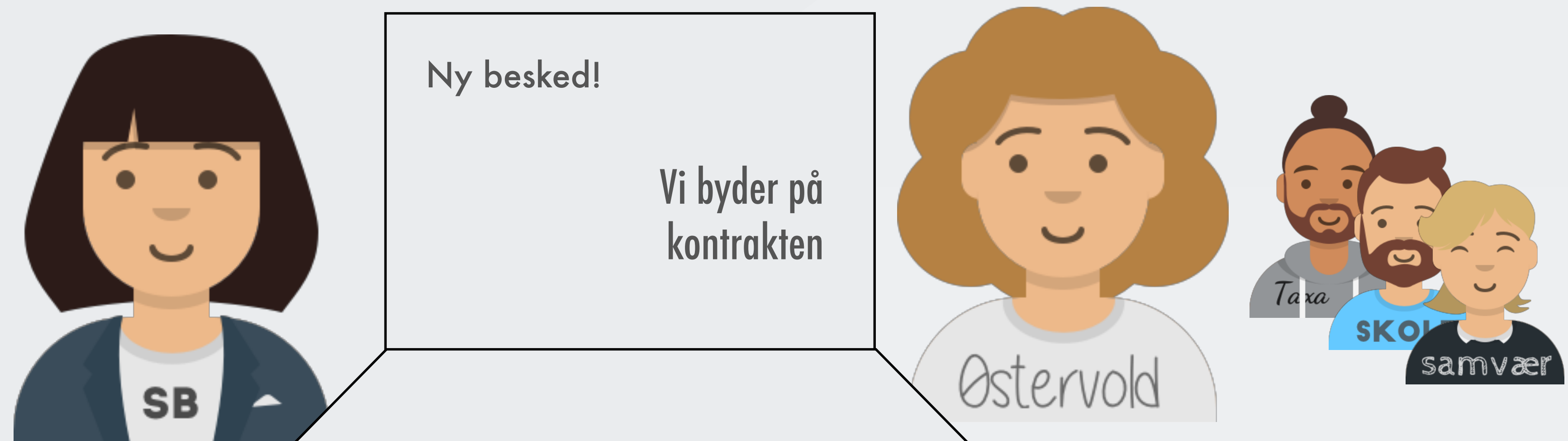
# DEL 1: KONTRAKTEN OPRETTES

SB tilpasser kontrakt-skabelon  
og tilføjer afhængige kontrakter.



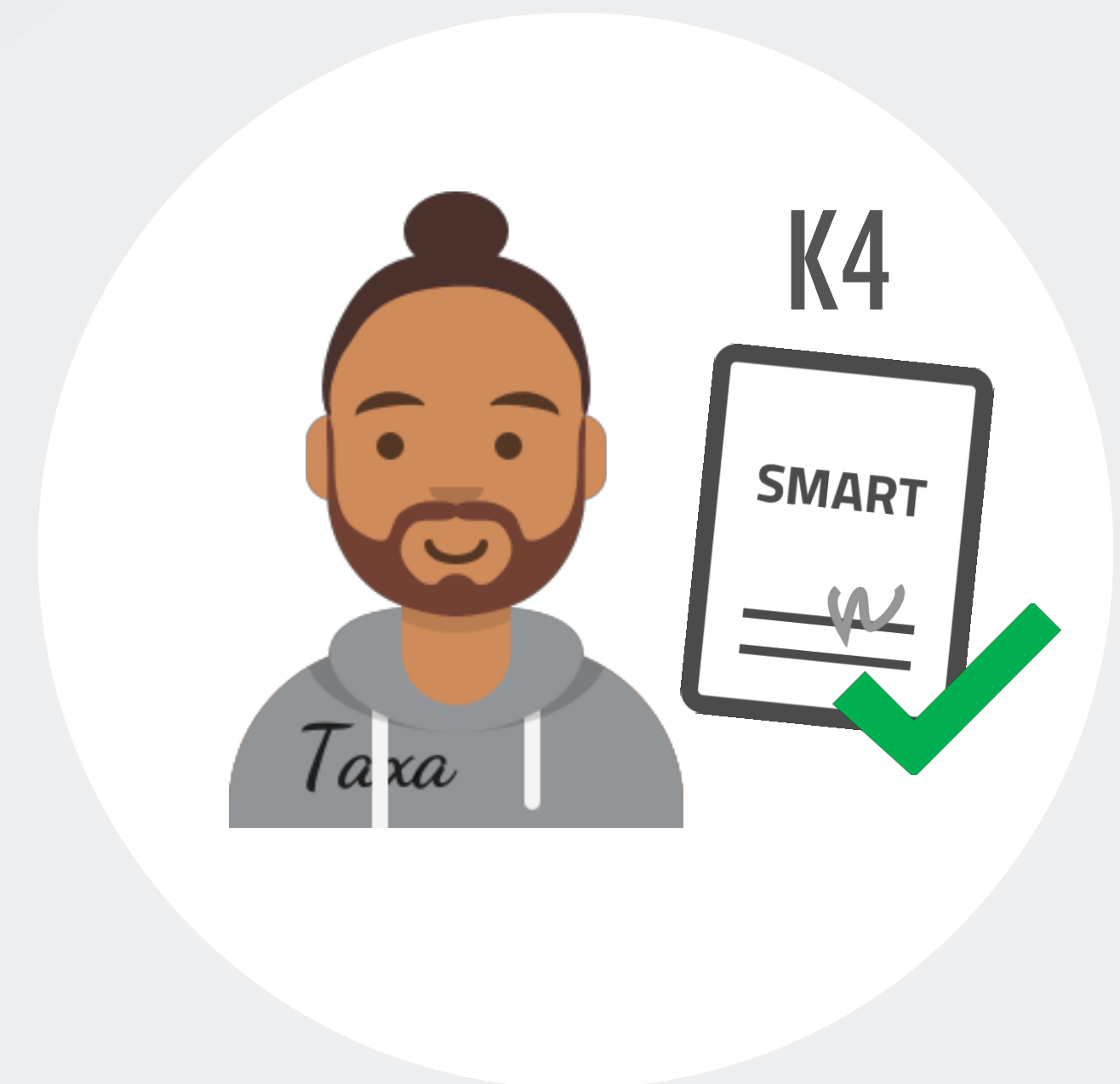
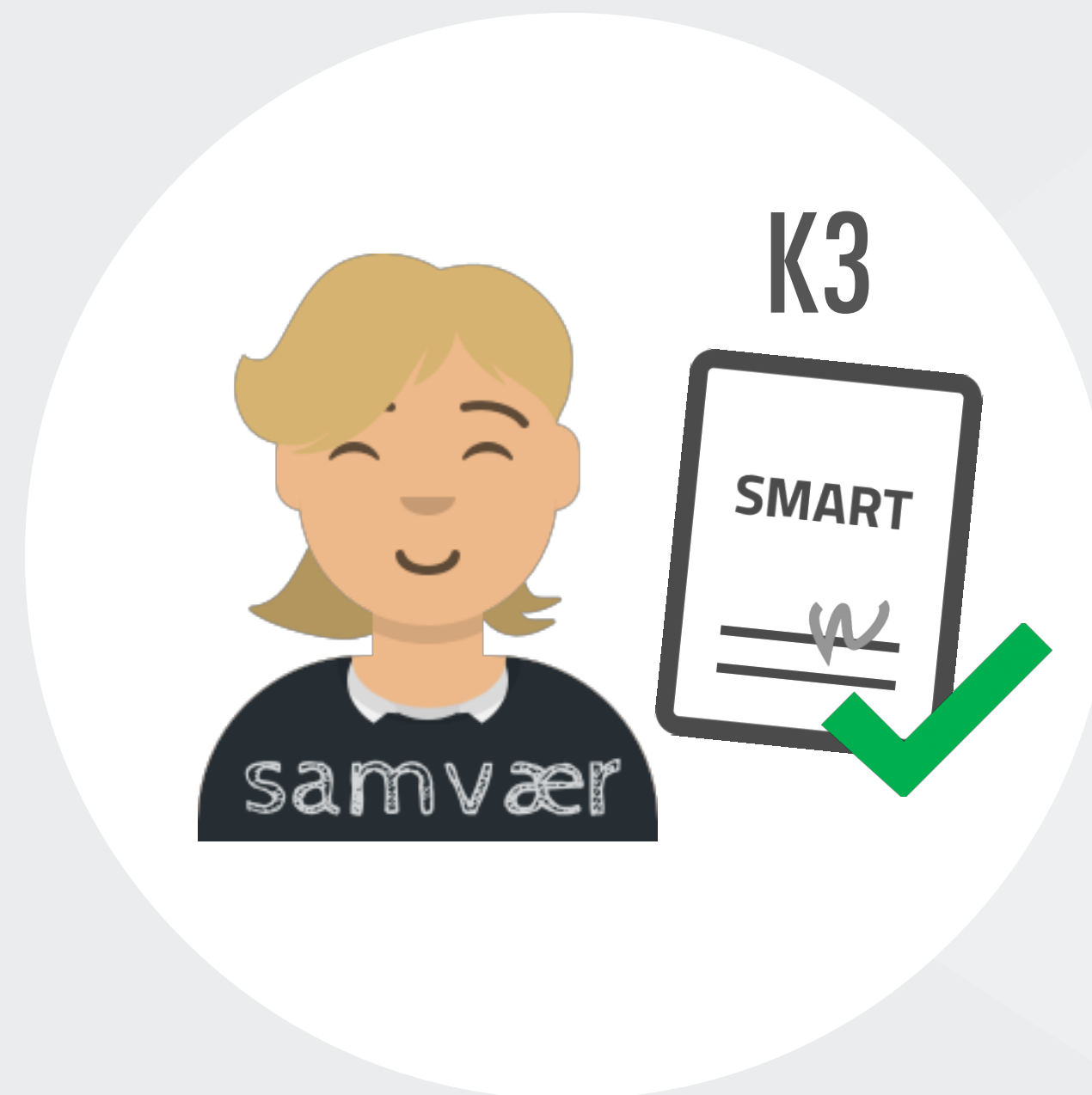
# DEL 1: KONTRAKTEN OPRETTES

SB modtager tilbud fra Østervold,  
samt tilbud på de afhængige kontrakter.

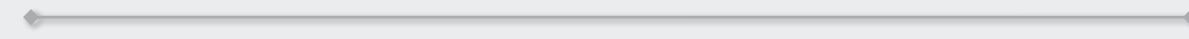


# DEL 1: KONTRAKTEN OPRETTES

Når alle aktører har godkendt,  
kan kontrakterne aktiveres.



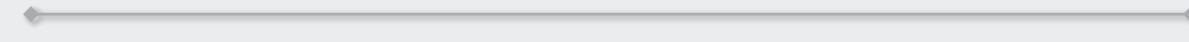
# DEL 1: KONTRAKTEN OPRETTES



Hvordan kunne brugergrænseflade se ud?



# DEL 1: KONTRAKTEN OPRETTES



**Hvordan fungerer det teknisk?**

# DEL 1: KONTRAKTEN OPRETTES

---

## Begrebsafklaring

Blockchain adresse

CVR nummer

Solidity

Kodesproget for kontrakterne

Smart kontrakt

Aftale + kontrakt + transaktioner

# DEL 1: KONTRAKTEN OPRETTES

1

**Start programmet**

```
$ npm run ganache
```

2

**Opret ny kontrakt kaldet "K1"**

```
$ node cli.js create K1
```

3

**Tilføj HTK og underleverandør af kontrakten**

```
<address_owner>
```

```
<address_serviceProvider>
```

# DEL 1: KONTRAKTEN OPRETTES

Din blockchain adresse er bevis for  
din identitet (ligesom CVR)

**HTKs blockchain adresse:**

0xea3fb1233e39bbd58448e19175576e228cc99077

**Leverandørs blockchain adresse:**

0xca5f61eac37ba5051d0e9305a32829923f7f0552

# DEL 1: KONTRAKTEN OPRETTES

4

**Tilføj de to "CVR-numre" i kontrakten.**

```
<0xea3fb1233e39bbd58448e19175576e228cc99077>  
<0xca5f61eac37ba5051d0e9305a32829923f7f0552>
```

5

**Leverandør og HTK godkender kontrakten.**

Notification:

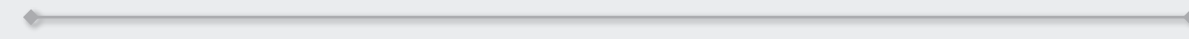
"HTK har godkendt kontrakten, og afventer jeres godkendelse for at aktivere"

6

**Aktivér kontrakten.**

```
K1.at('0xCA1C7caC964471A87e621A9297598c1ce').activate()
```

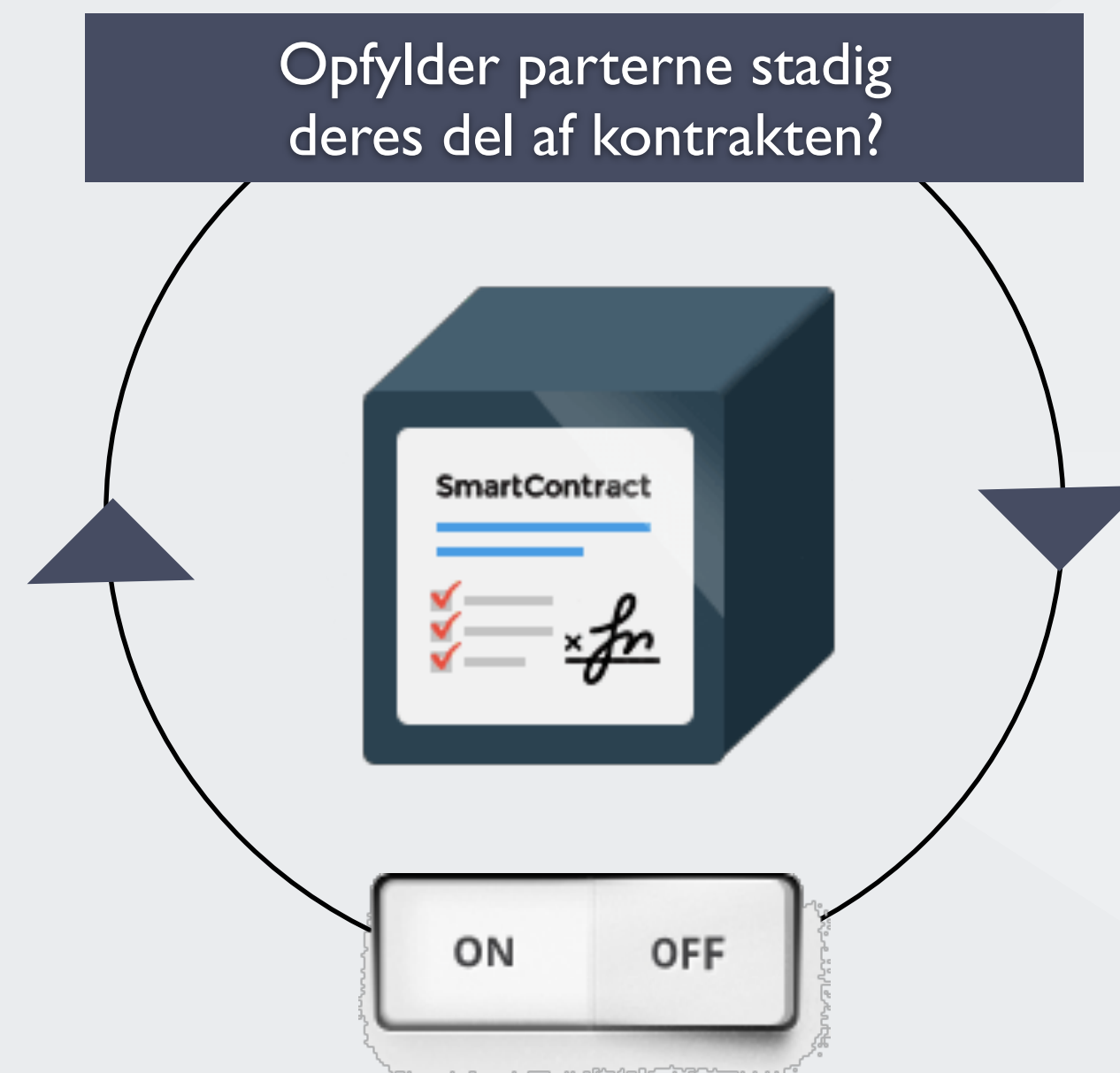
# DEL 2: KONTRAKTEN ER AKTIV



**Koncept illustreret**

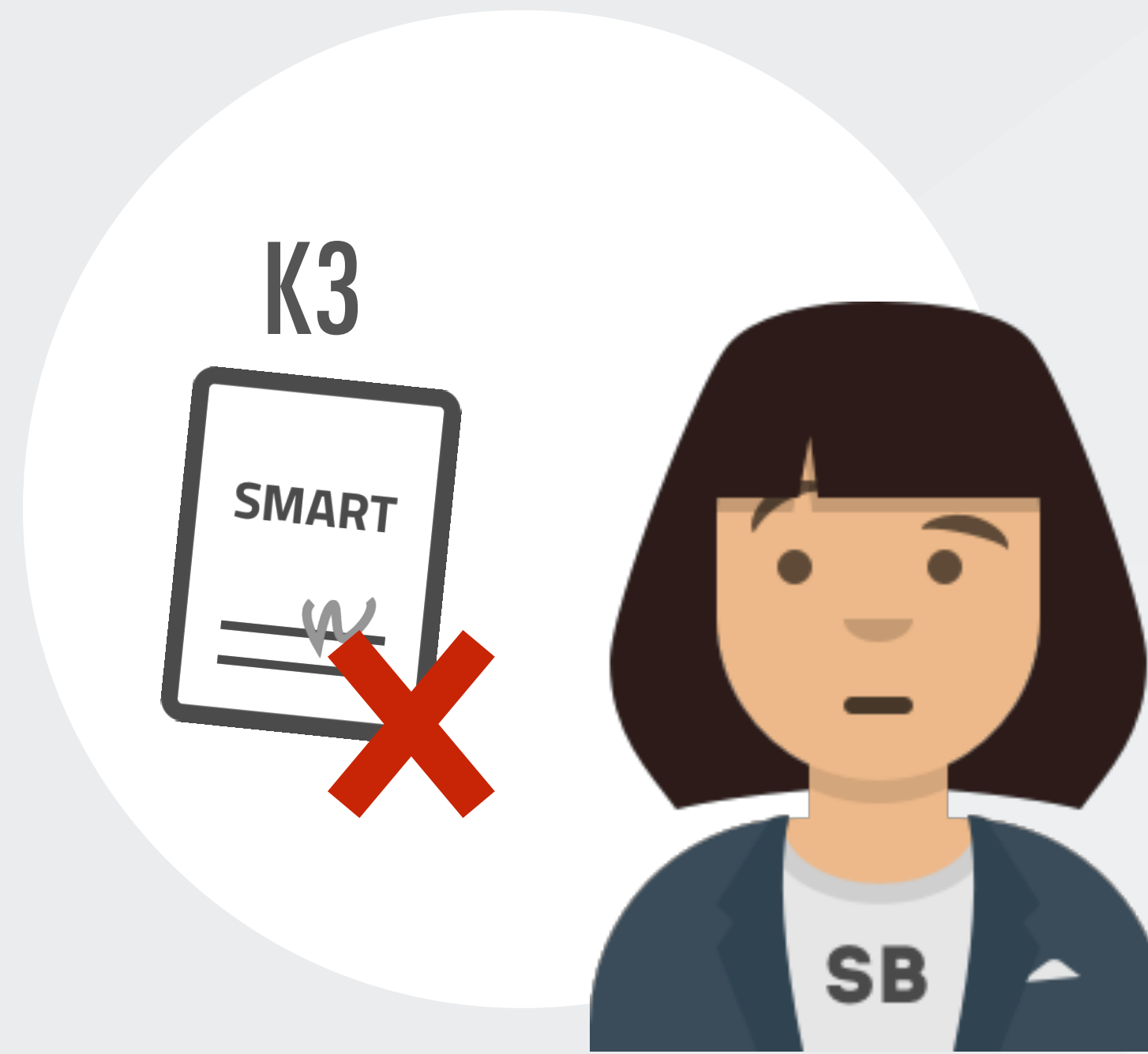
## DEL 2: KONTRAKTEN ER AKTIV

Her begynder det "smarte":  
Kontrakterne tjekker sig selv.



## DEL 2: KONTRAKTEN ER AKTIV

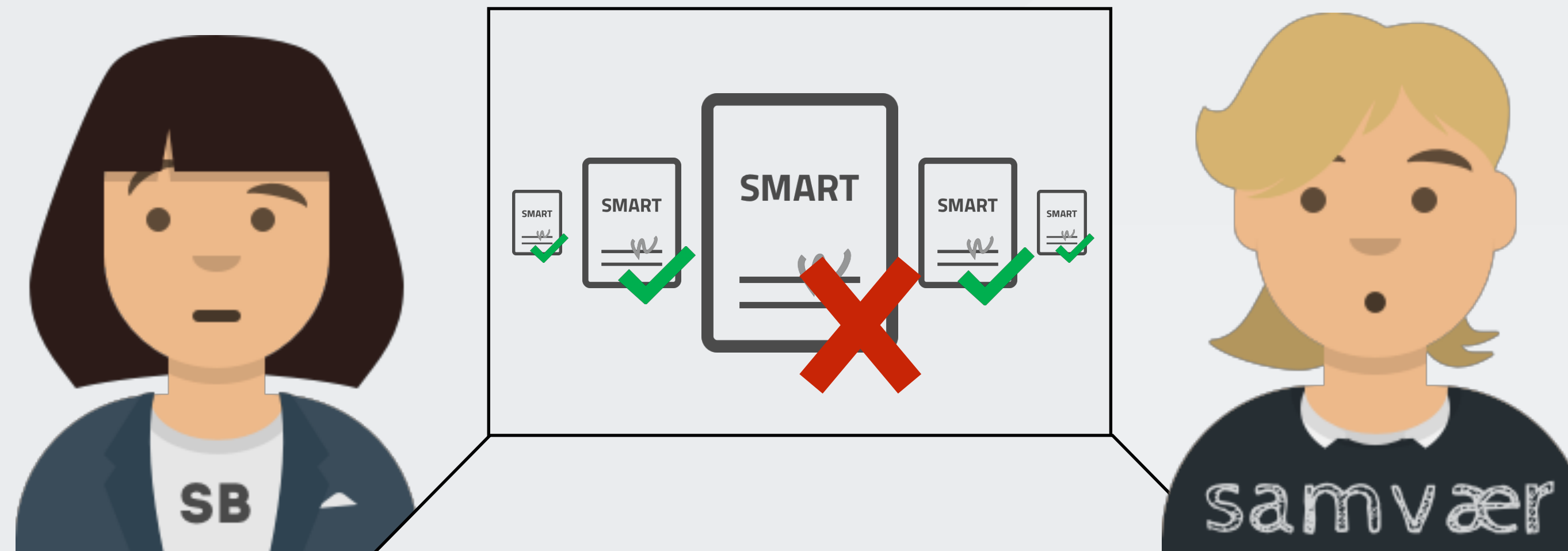
Kontrakten registrerer et problem:  
Nyeste status mangler





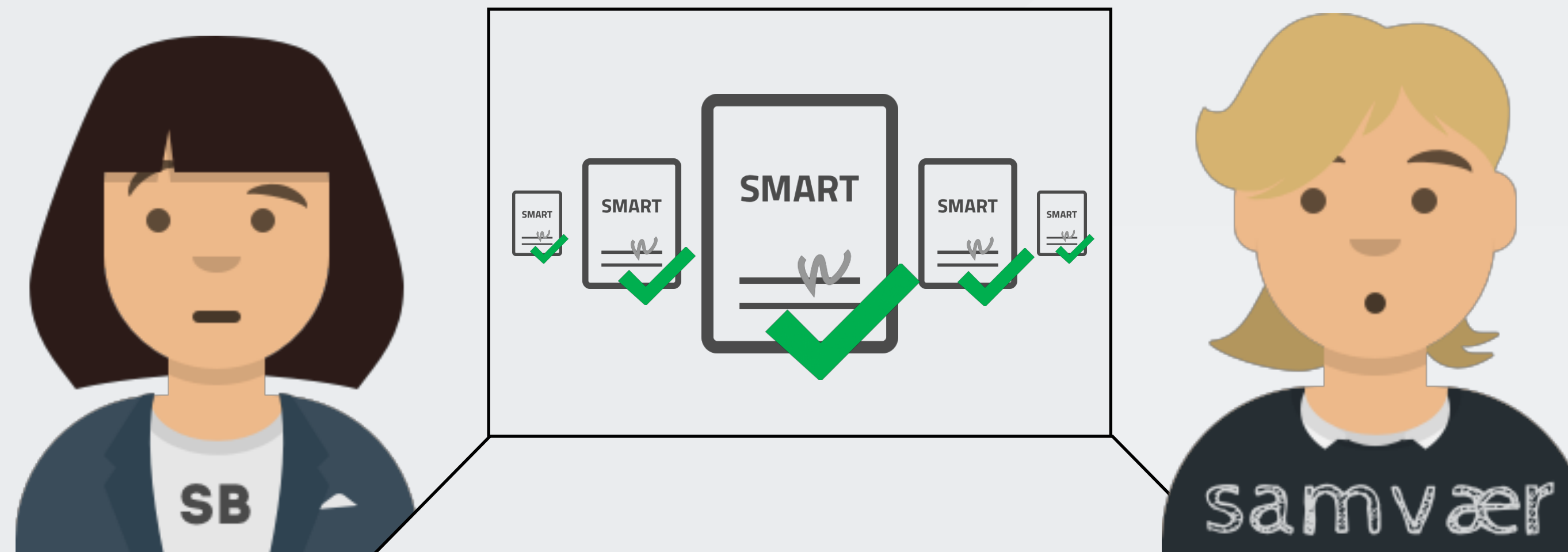
## DEL 2: KONTRAKTEN ER AKTIV

SB ringer til underleverandør,  
for at høre om det er en fejl.



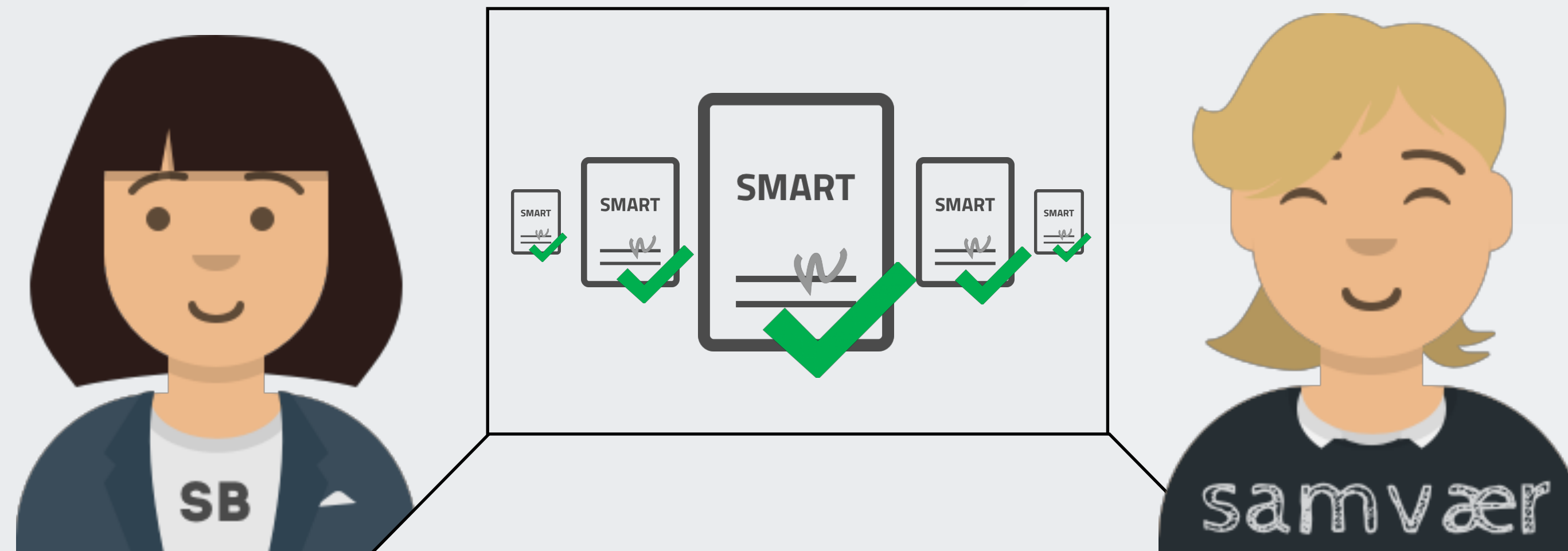
## DEL 2: KONTRAKTEN ER AKTIV

Underleverandør uploader  
den manglende status.

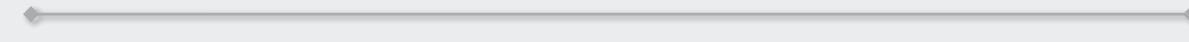


## DEL 2: KONTRAKTEN ER AKTIV

Smart-kontrakten registrerer det,  
og fortsætter udbetaling som hidtil.

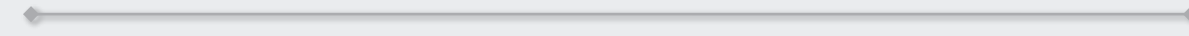


## DEL 2: KONTRAKTEN ER AKTIV



Hvordan kunne brugergrænseflade se ud?

# DEL 2: KONTRAKTEN ER AKTIV



**Hvordan fungerer det teknisk?**

## DEL 2: KONTRAKTEN ER AKTIV

1

**Tjek kontrakt, og betal hvis kontrakt er aktiv**  
`$ node cli.js pay`

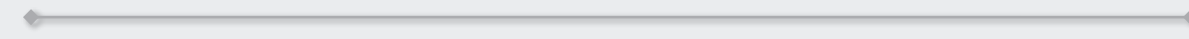
2

**Sæt transaktioner i bero, indtil kontrakt er aktiv**  
`$ node cli.js set status`

3

**Start overførsler til kontrakt igen**  
`$ node cli.js pay`

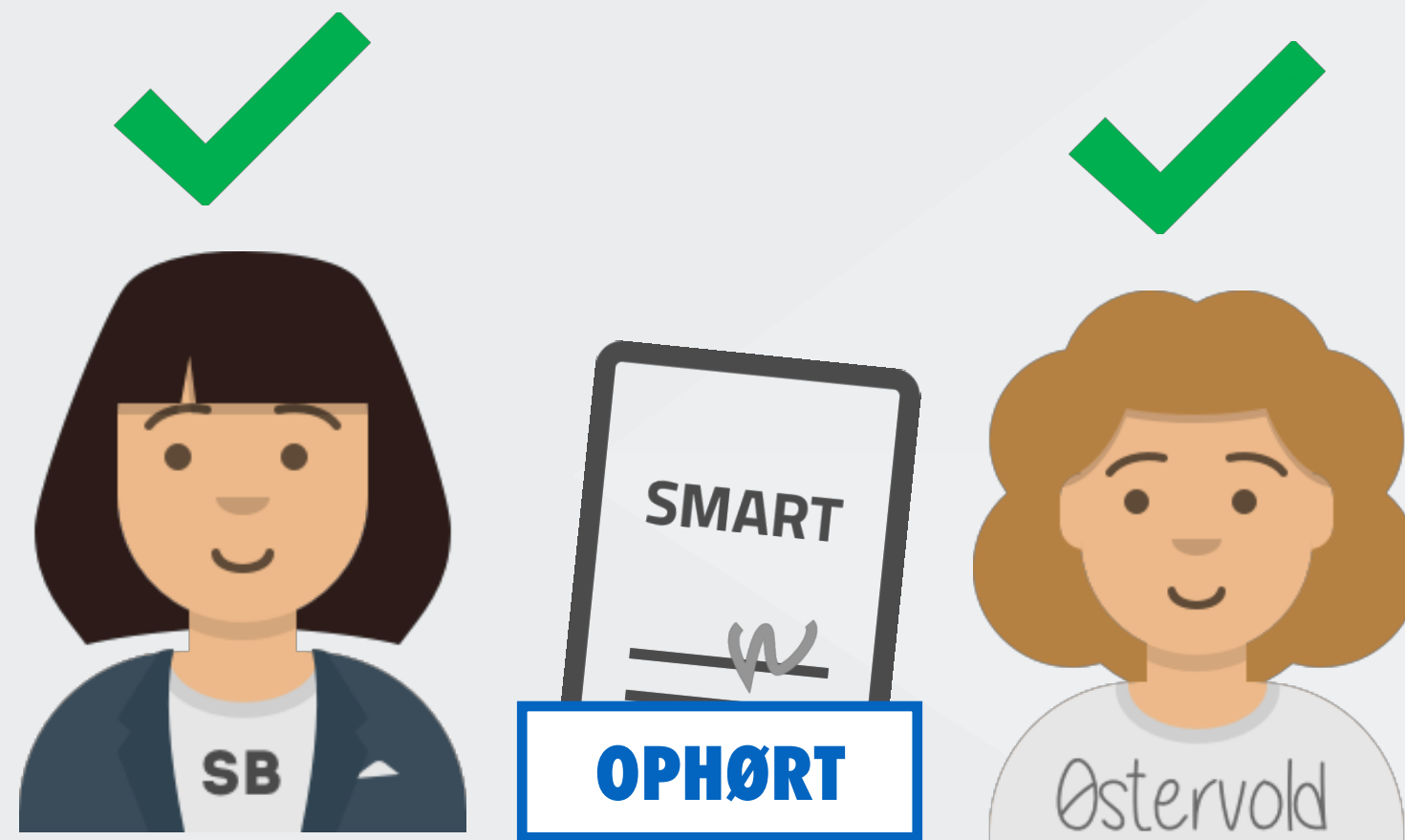
# DEL 3: KONTRAKTEN OPHØRER



**Koncept illustreret**

# DEL 3: KONTRAKTEN OPHØRER

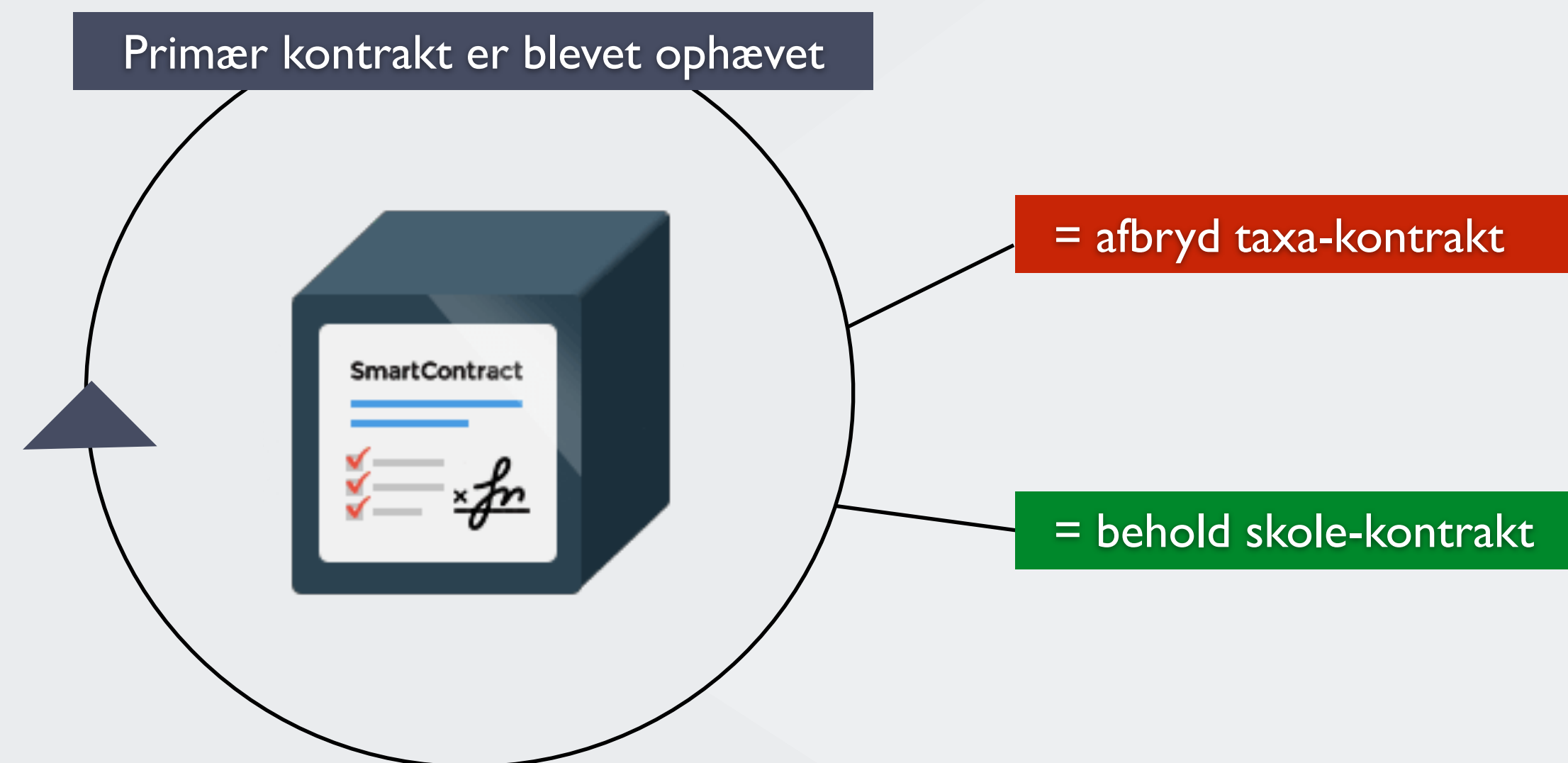
SB beslutter at Preben skal hjemgives,  
så kontrakt med bostedet ophører.



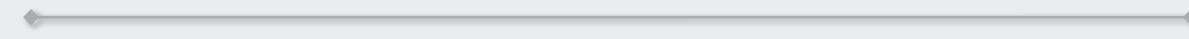


# DEL 3: KONTRAKTEN OPHØRER

SB beslutter at Preben skal hjemgives,  
så kontrakt med bostedet ophører.



# DEL 3: KONTRAKTEN OPHØRER



Hvordan kunne brugergrænseflade se ud?

# DEL 3: KONTRAKTEN OPHØRER

---

Hvordan fungerer det teknisk?

# DEL 3: KONTRAKTEN OPHØRER

1

**Tjek status**

```
$ node cli.js pay
```

2

**Kontrakten er udløbet**

```
subContract.getState() == Active
```

3

**Kontrakten med taxa-selskabet stoppes**

```
state = Expired
```

# OPSUMMERING

På pilot projektet

## Nuværende situation

- Høj kompleksitet i aftaler
- Ressourcekrævende at validere kontrakter
- Store variationer i kontrakter
- Ingen "serve-ret" på aftaler

## Blockchain løsningen

- Et fælles kontrakt-overblik
- Standardisering af kontrakter
- Sikrer korrekt udbetaling
- Automatisk validering af fakturaer

## Fordelene/resultatet

- Større sikkerhed for alle parter
- Reducerer administrativ arbejdsbyrde, så ressourcer kan gå til fagligt arbejde

# NEXT STEPS

---

## Udvid Scope:

- Flere kommuner
- Udfold værdi: andre cases
- Inddrag flere problemstillinger
- Undersøg feasibility
- Uddyb eventuelle udfordringer

# TAK FOR SAMARBEJDET

---



**KOMBIT**

**KL**