Stratis

STRATIS

We make blockchain easy for you.

"Smart Contracts as Rules for Money - A case study"

Rowan de Haas

Background

Long-time .NET developer

Involved in Ethereum / solidity smart contract development

Working on Stratis' C# smart contracts platform

Stratis **Platform**

Stratis is a cryptocurrency (STRAT)

Stratis is an open-source platform

Stratis is a consultancy

We have expertise from our public cryptocurrency

We put that expertise into our technology platform

We consult with enterprises to build solutions around our platform

Stratis

Stratis **Smart Contracts**

Stratis

Smart **Contracts**

Lots of financial instruments are rules for money

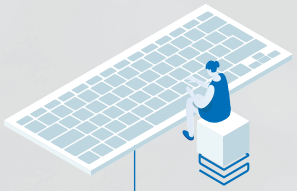
Lots of business processes are rules for money

Lots of legislation is rules for money

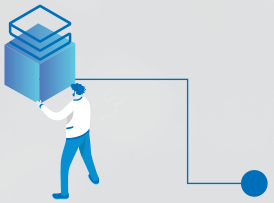
Stratis

Smart **Contracts**

An Australian project management firm recognised this



Let's look at their case study



Worked with Stratis to implement a trial

Trade payments contract

Based on a piece of Australian legislation

Stratis

Case Study: **Project Manager**

Wildara

ISO9001 certified project managers

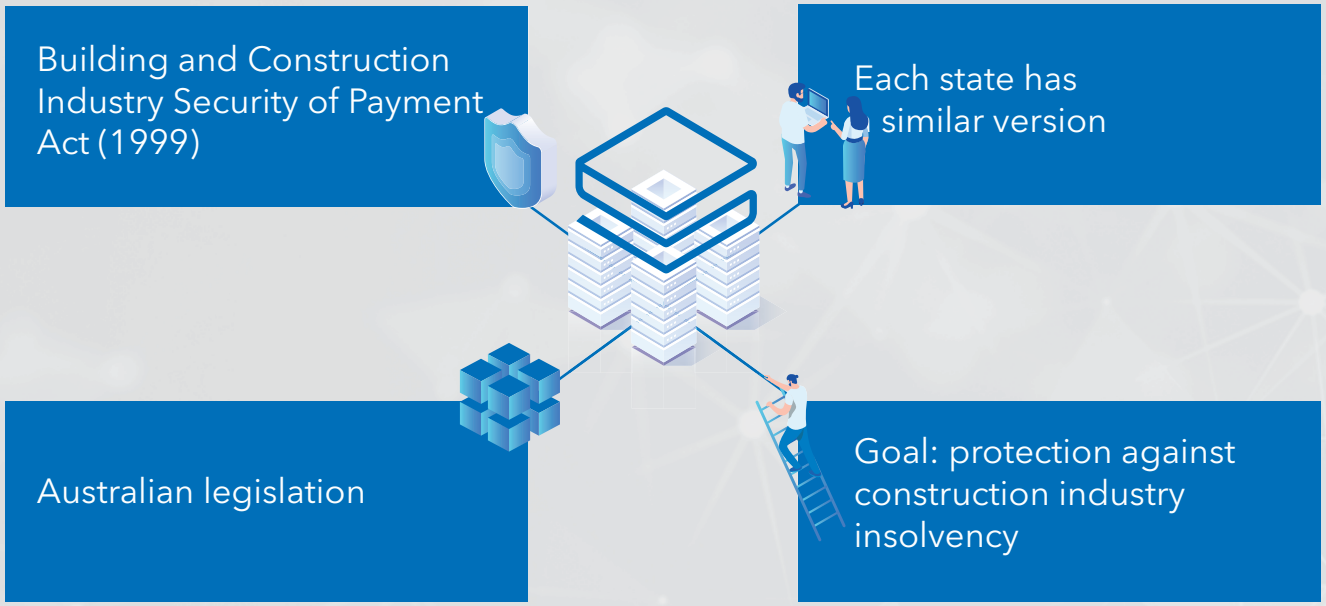
Clients:
- Australian government
- Department of Defence
- Local government
- Commercial and retail

Case Study: **Contractor**

- Diprotodon
- Earthworks contractor
- Excavation
- Earthmoving
- Site preparation

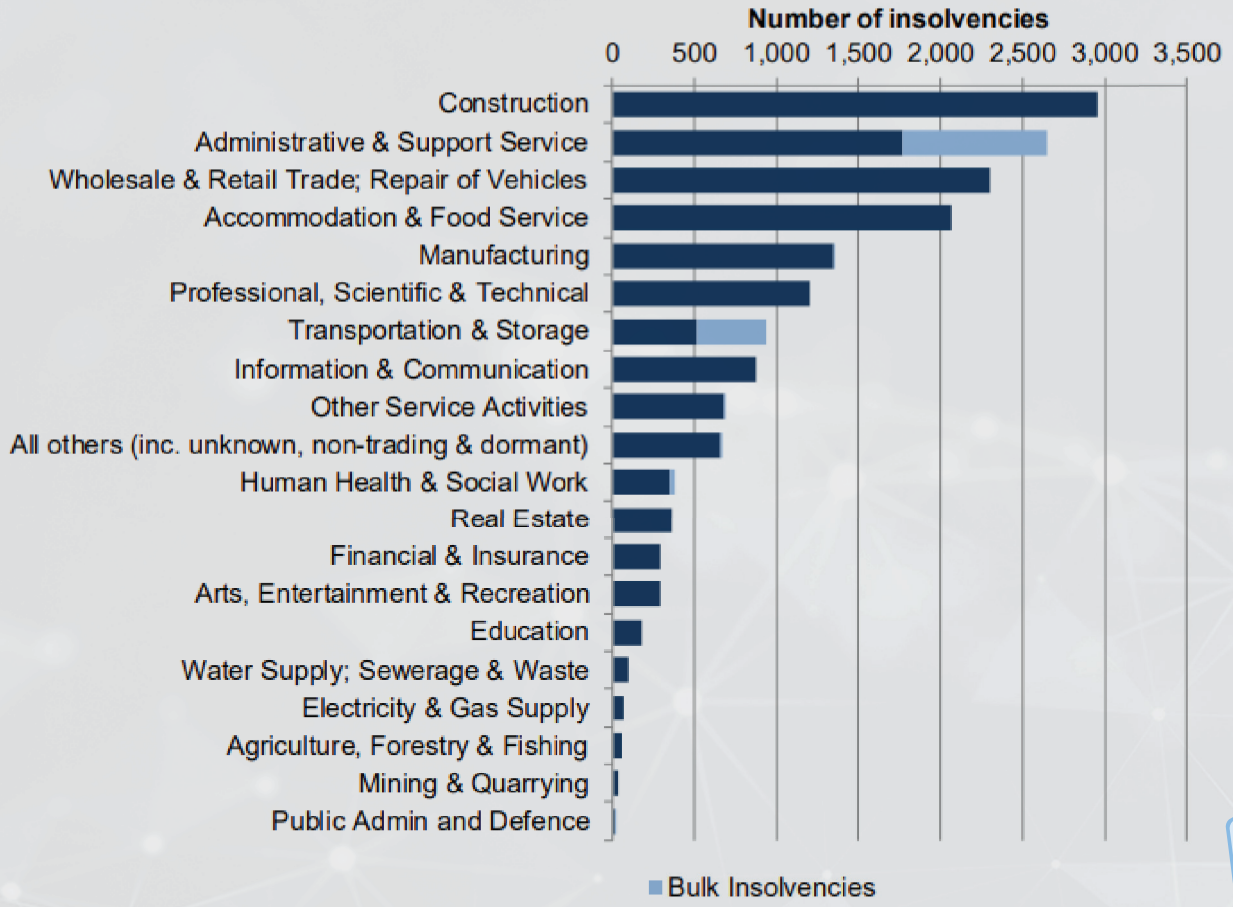
Stratis

Case Study:
Security of Payment Act



Stratis

Total new company insolvencies in England and Wales by broad industry grouping, 12 months ending Q4 2018



Source: Insolvency Service and Companies House.

Image source: Insolvency Service – Official Statistics Q4 2018

Stratis

Case Study:
Security of Payment Act

Contractors have legal right to progress payments for work delivered

These happen on a regular basis

The act provides protection

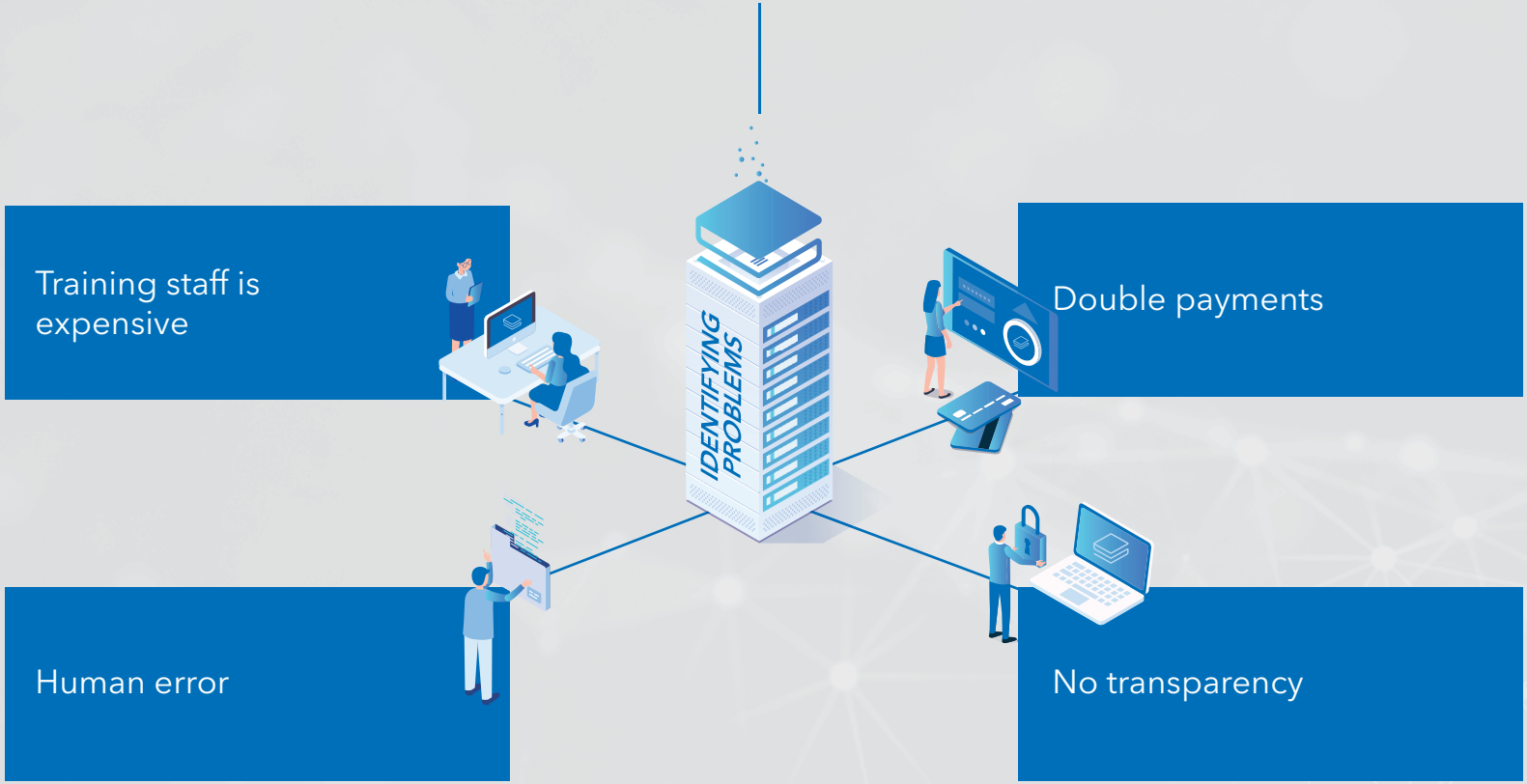
- Short maximum payment terms
- Minimum interest rates
- Dispute resolution flow

Maximum time to get payment: 15 days

Rules for money

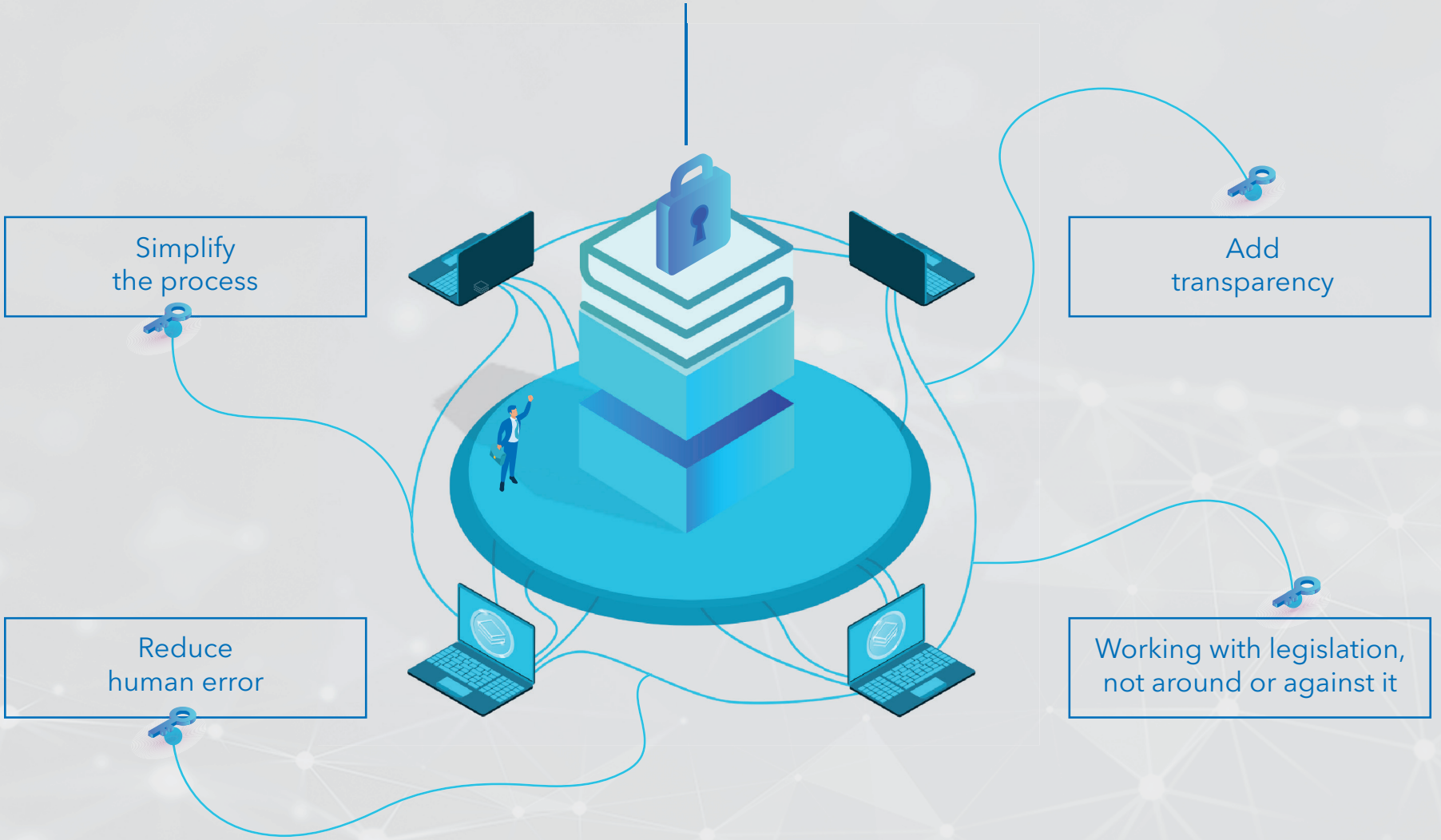
Stratis

Case Study:
Identifying Problems



Stratis

Case Study:
Goals

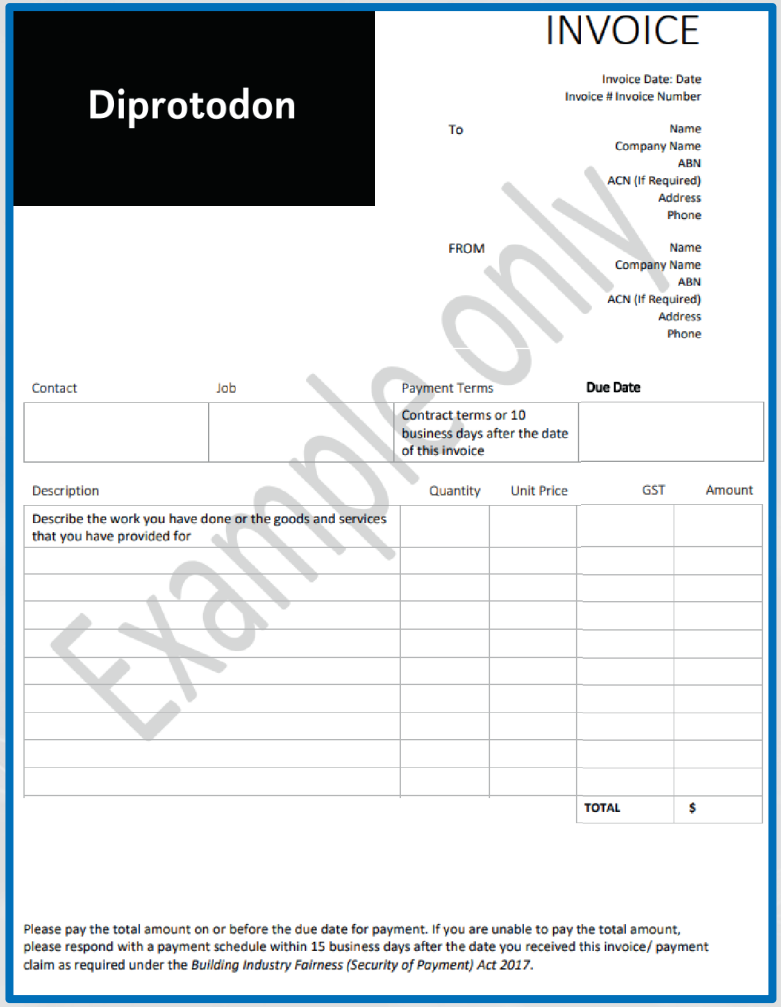


Stratis

Case Study:
Security of Payment Act

Stratis

Step 1:
Contractor Issues Payment Claim



Diprotodon

INVOICE

Invoice Date: Date
Invoice # Invoice Number

To
Name
Company Name
ABN
ACN (If Required)
Address
Phone

FROM
Name
Company Name
ABN
ACN (If Required)
Address
Phone

Contact	Job	Payment Terms	Due Date
		Contract terms or 10 business days after the date of this invoice	

Description	Quantity	Unit Price	GST	Amount
Describe the work you have done or the goods and services that you have provided for				
			TOTAL	$

Please pay the total amount on or before the due date for payment. If you are unable to pay the total amount, please respond with a payment schedule within 15 business days after the date you received this invoice/ payment claim as required under the *Building Industry Fairness (Security of Payment) Act 2017*.

Example only

```
public class PaymentContract
{
  number ClaimAmount = 1000;

  number Balance = 0;

  bool Paid = false;

  Address Admin = "Diprotodon";

  void PayClaim() {}

  void Withdraw(ulong amount) {
    Assert(Sender == Admin);
    Transfer(Admin, Balance);
  }
}
```

Stratis

Step 2:
Project Manager Agrees With Claim

PAYMENT SCHEDULE

TO (CLAIMANT'S NAME): _____

ABN or ACN (where applicable):

_____

Address (ordinary place of business): _____

_____

Phone Number: _____ Fax Number: _____

FROM (RESPONDENT'S NAME): _____

ABN or ACN (where applicable):

_____

Address (ordinary place of business): _____

_____

Phone Number: _____ Fax Number: _____

CONTRACT DETAILS:

Project: _____

Contract Number (where applicable): _____

Date on payment claim and claim number, if any: _____

Date when the Payment Claim was served on the respondent: _____

Total amount of the Payment Claim: $_____

Amount that the respondent proposes to pay (the 'scheduled amount'): $ _____

If the scheduled amount is less than the claimed amount, the reasons why it is less and the reasons for withholding payment are set out in the Attachment(s) below:

Signed (Respondent): _____

Date: _____

ATTACHMENT(S)

Details of Payment Schedule (attach other relevant documentation as required):

This is a payment schedule made under the Building and Construction Industry Security of Payment Act 2002 (Vic).

WILDARA™

```
public class PaymentContract
{
  number ClaimAmount = 1000;

  number Balance = 0;

  bool Paid = false;

  Address Admin = "Diprotodon";

  void PayClaim() {}

  void Withdraw(ulong amount) {
    Assert(Sender == Admin);
    Transfer(Admin, Balance);
  }
}
```

Stratis

Step 2:
Project Manager Agrees With Claim

PAYMENT SCHEDULE

TO (CLAIMANT'S NAME): _____

ABN or ACN (where applicable):

_____

Address (ordinary place of business): _____

_____

Phone Number: _____ Fax Number: _____

FROM (RESPONDENT'S NAME): _____

ABN or ACN (where applicable):

_____

Address (ordinary place of business): _____

_____

Phone Number: _____ Fax Number: _____

CONTRACT DETAILS:

Project: _____

Contract Number (where applicable): _____

Date on payment claim and claim number, if any: _____

Date when the Payment Claim was served on the respondent: _____

Total amount of the Payment Claim: $_____

Amount that the respondent proposes to pay (the 'scheduled amount'): $ _____

If the scheduled amount is less than the claimed amount, the reasons why it is less and the reasons for withholding payment are set out in the Attachment(s) below:

Signed (Respondent): _____

Date: _____

ATTACHMENT(S)

Details of Payment Schedule (attach other relevant documentation as required):

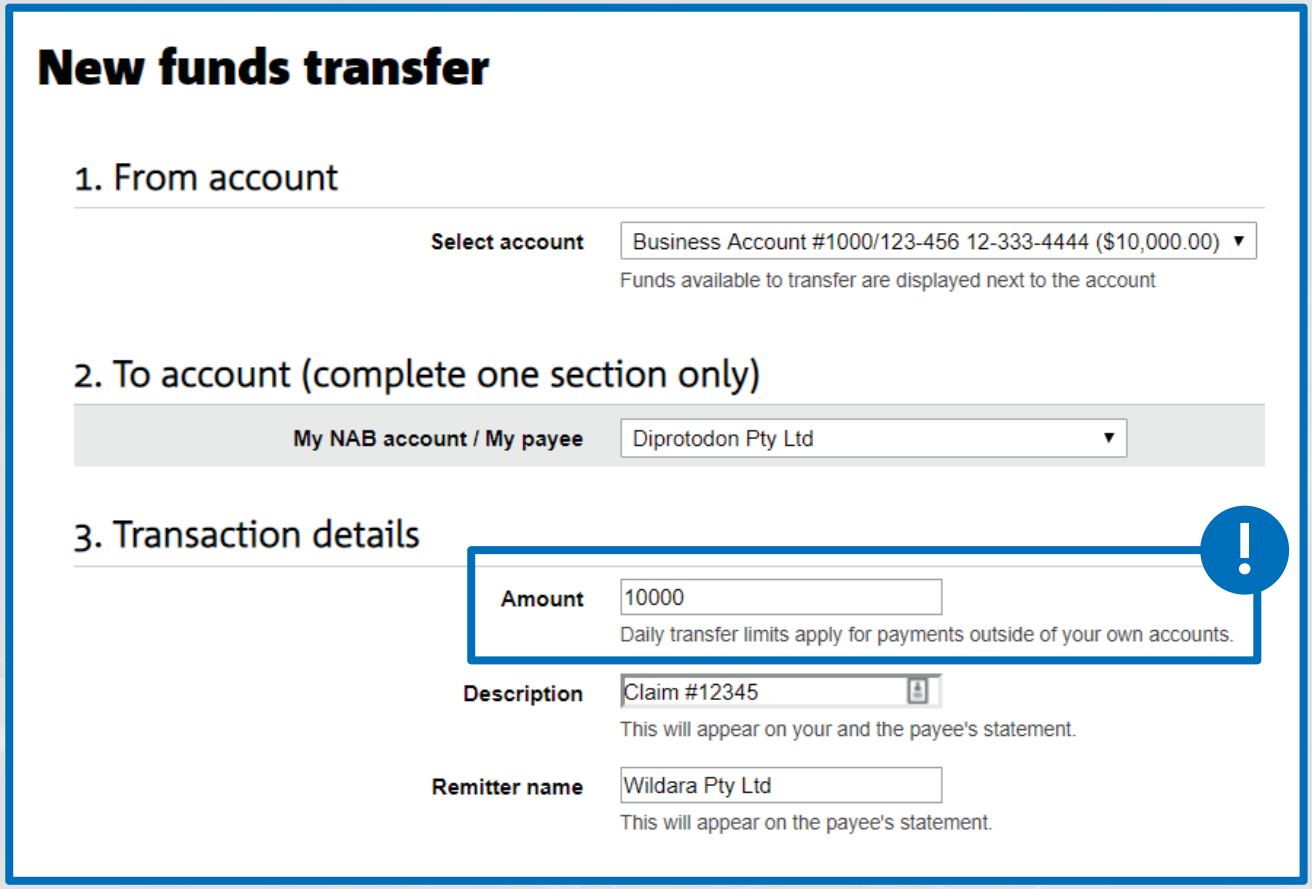
This is a payment schedule made under the Building and Construction Industry Security of Payment Act 2002 (Vic).

WILDARA™

```
public class PaymentContract
{
  number ClaimAmount = 1000;

  number Balance = 1000;

  bool Paid = true;

  Address Admin = "Diprotodon";

  void PayClaim() {}

  void Withdraw(ulong amount) {
    Assert(Sender == Admin);
    Transfer(Admin, Balance);
  }
}
```

Stratis

Step 3:
Project Manager Pays Claim




```
public class PaymentContract
{
  number ClaimAmount = 1000;

  number Balance = 1000;

  bool Paid = true;

  Address Admin = "Diprotodon";

  void PayClaim() {}

  void Withdraw(ulong amount) {
    Assert(Sender == Admin);
    Transfer(Admin, Balance);
  }
}
```

Stratis

Step 4:
Contractor withdraws funds

+1000

```
public class PaymentContract
{
  number ClaimAmount = 1000;

  number Balance = 0;

  bool Paid = true;

  Address Admin = "Diprotodon";

  void PayClaim() {}

  void Withdraw(ulong amount) {
    Assert(Sender == Admin);
    Transfer(Admin, Balance);
  }
}
```

Stratis

Case Study: **Implementation**

- C# Smart contract
- Running on Cirrus sidechain
- Purchase STRAT
- Send to Wildara wallet
- Execute contract on site in Wildara office
- Withdraw to Diprotodon wallet
- Sell STRAT

```
public class SOPPaymentContract : SmartContract
{
    public SOPPaymentContract(ISmartContractState smartContractState, ulong claimAmount)
    : base(smartContractState)
    {
        Assert(claimAmount > 0);

        AdminAddress = Message.Sender;
        ClaimAmount = claimAmount;
        PaidAmount = 0;
    }

    public ulong PaidAmount
    {
        get => PersistentState.GetUInt64(nameof(PaidAmount));
        private set => PersistentState.SetUInt64(nameof(PaidAmount), value);
    }

    public Address AdminAddress
    {
        get => PersistentState.GetAddress(nameof(AdminAddress));
        private set => PersistentState.SetAddress(nameof(AdminAddress), value);
    }

    public ulong ClaimAmount
    {
        get => PersistentState.GetUInt64(nameof(ClaimAmount));
        private set => PersistentState.SetUInt64(nameof(ClaimAmount), value);
    }

    public void Withdraw(ulong amount)
    {
        Assert(Message.Sender == AdminAddress);
        Assert(Balance >= amount);

        Transfer(Message.Sender, amount);
    }

    public void Deposit()
    {
        Assert(PaidAmount + Message.Value <= ClaimAmount);
        PaidAmount += Message.Value;
    }
}
```

Stratis
COUDE 10·12
MANCHON 28·NOIR
TES 26·28
RACCORD N·12·20·22 CB
RACCORD N·12·26·28 CB
COLLIE 28
TES 14·16
COUDE 12·14
COLLIER

Stratis

Lessons

Ease-of-use
Improved contract interaction through a DApp

Currency
Fiat peg (Australian dollar)

Stratis

The **Future**

Smart contracts as business logic

Legislation and smart contracts working together

Stratis

THANK YOU

rowandehaas

DISCORD

https://discord.gg/9tDyfZs

@rowandehaas
@stratisplatform