



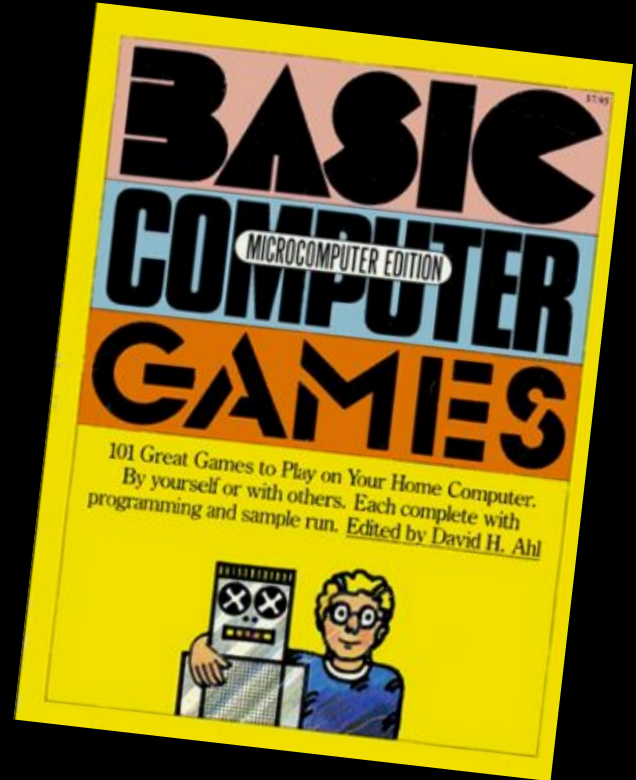
Opening Up Financial Services to Grow the SA Tech Ecosystem

Wayne Summers | Head of Digital IT



I'm Wayne Summers 🖐️

- Self taught developer
- Mainframes
- Developer in a banker's world
- Excited about the possibilities of tech



Why am I here?



“ We believe that we can make a difference in growing the SA economy by growing the technology sector.



“ We believe that we can make a difference in growing the SA economy by growing the technology sector.

growing the technology sector

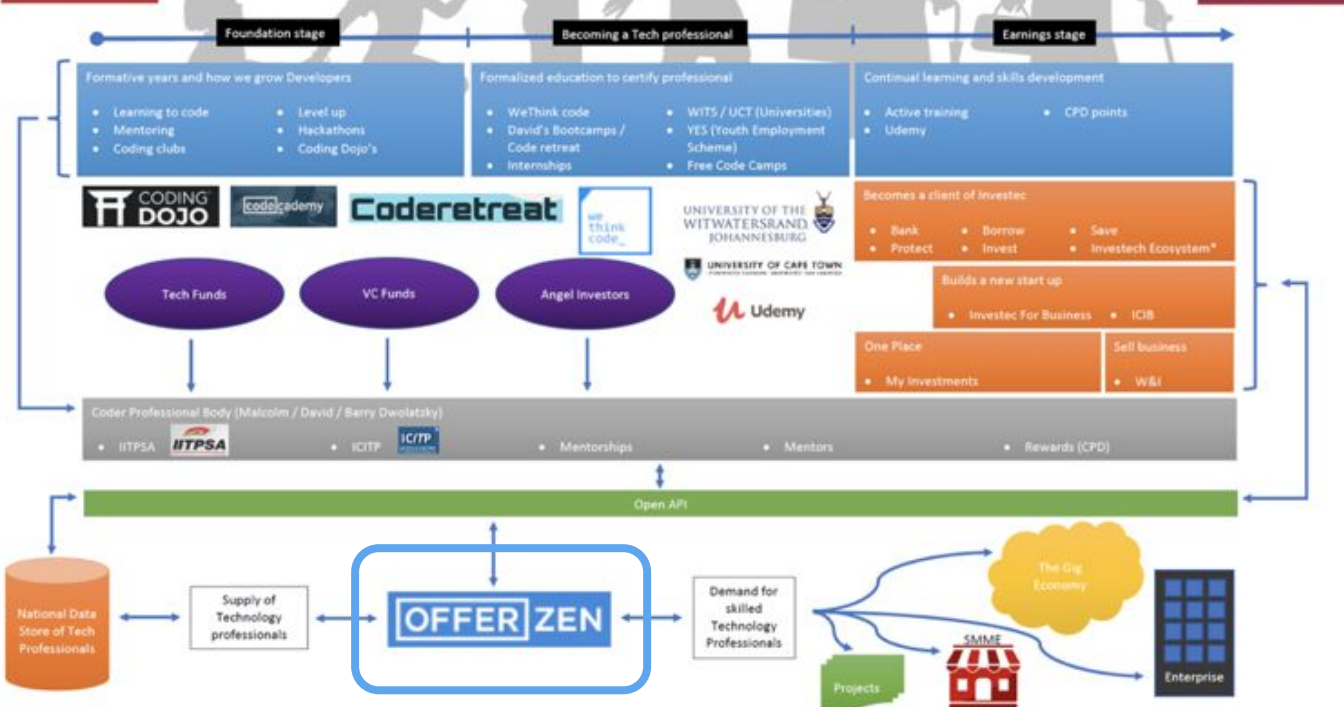


*more devs making
more awesome software*



How do we empower
SA's devs?

Tech Ecosystem





Connection lost

Connection with your device
lost. Try reconnecting again.
chromecast.com/help

© 2015 Google Inc.

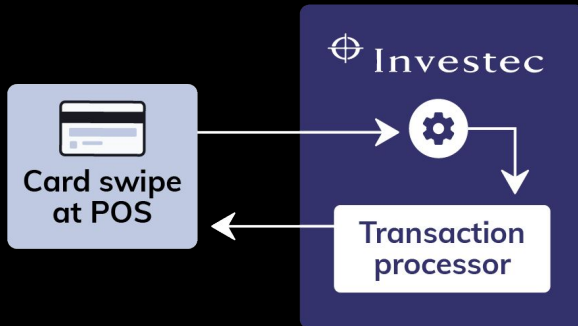


As a bank, the most impactful thing
we can do for devs is to enable
them to programmatically access
their money and data.

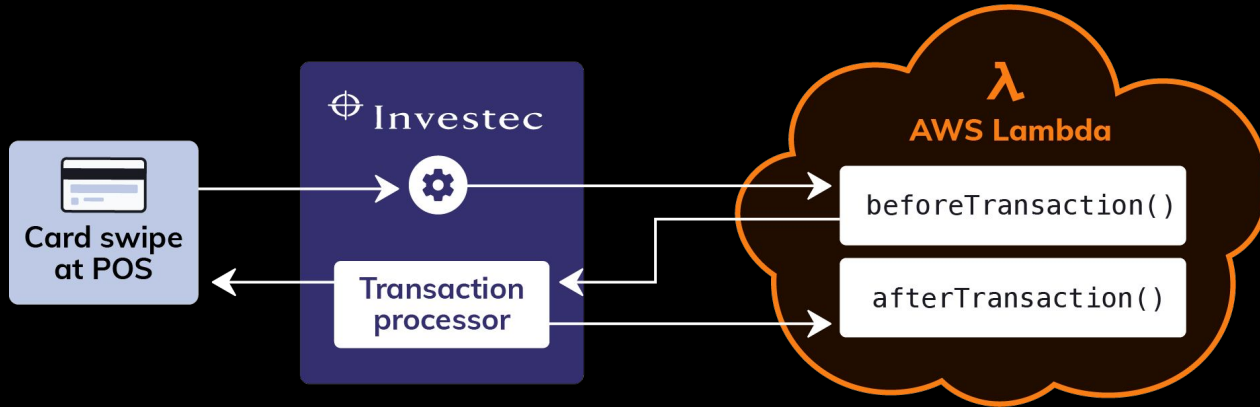


MVP Experiment


How cards work






Levelling up Investec cards - thanks Root ;)




A hosted code editor




Notifications (1)  Logout  Menu 

Simulator ▾402167xxxxx8016Back

Cents amount 

Currency

 South African Rand ▾

Merchant category

Grocery Stores, Supermarkets ▾

Merchant name

Spar

Merchant city

Cape Town

Run in Sandbox

Console

Before transaction

After transaction

No output yet, [run a test](#)

Script is: **Active** | Published

DeactivatePublish

```
1 // This function runs during the card transaction authorization flow.
2 // It has a limited execution time, so keep any code short-running.
3 const beforeTransaction = async (authorization) => {
4   /** Example declining a transaction */
5   // if (authorization.merchant.category.key === 'fast_food_restaurants') {
6   //   return false; // Decline the transaction
7   // }
8
9   console.log(authorization)
10
11   // Convert rands to cents
12   const randAmount = authorization.centsAmount / 100;
13   // If my transaction category is equal to 'Travel Agencies, Tour Operators' or less than R500
14   if (authorization.merchant.category.key === 'Travel Agencies, Tour Operators' || randAmount < 500) {
15     return true; // Approve the transaction
16   } else {
17     // Send myself a sms reminding us why it was declined
18     const sms = await notifications.sms({
19       mobile: '+27831234567',
20       body: 'Fraudulent transaction'
21     })
22     sms.send();
23     return false; // Decline transaction
24   }
25 }
26
27
28 // This function runs after a transaction was successful.
29 const afterTransaction = async (transaction) => {
30   /** Example notifications */
31   // const amount = formatCurrency(authorization.centsAmount, authorization.currencyCode);
32   // const message = amount + ' spent at ' + authorization.merchant.name;
33   // notifications.sendSMS(message, '+27821234567');
34   // notifications.sendEmail(message, 'example@example.com');
35   // notifications.sendPush(message);
36 }
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Simple use case demo



1. I want to budget and track my spend

2. I want to get notified of my remaining limit

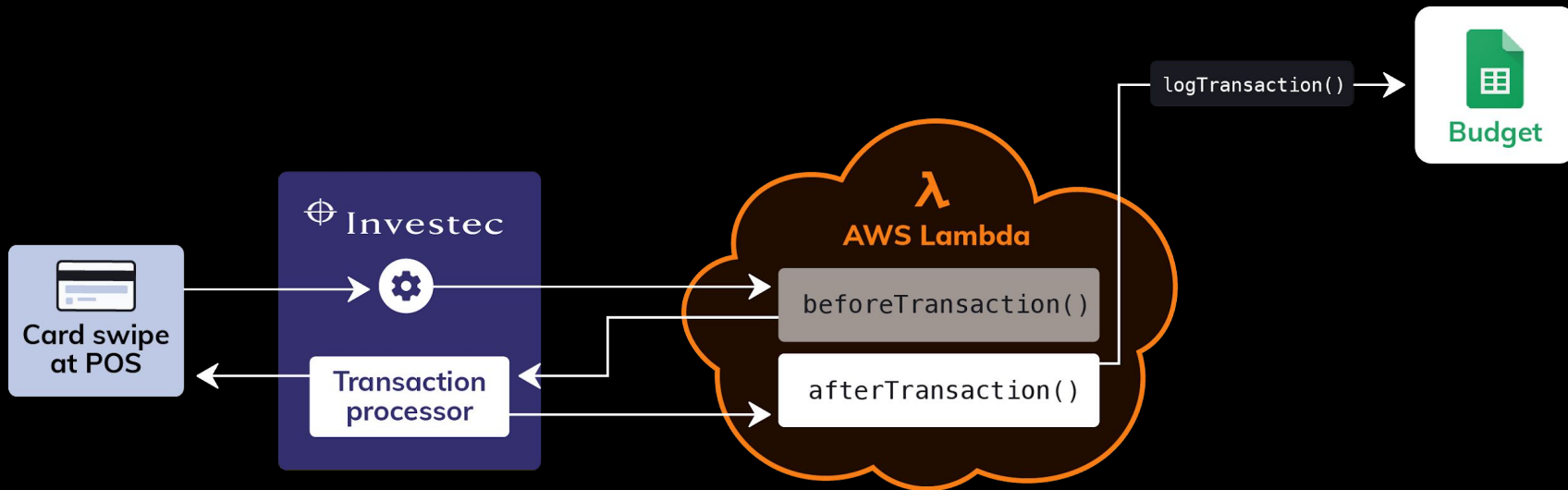
3. I want to stop myself from going over budget



1. I want to budget and track
my spend

Pushing transaction data into Google Sheets using

`afterTransaction()`



1	Weekly Budget
---	----------------------

2	Summary	Spending
3		

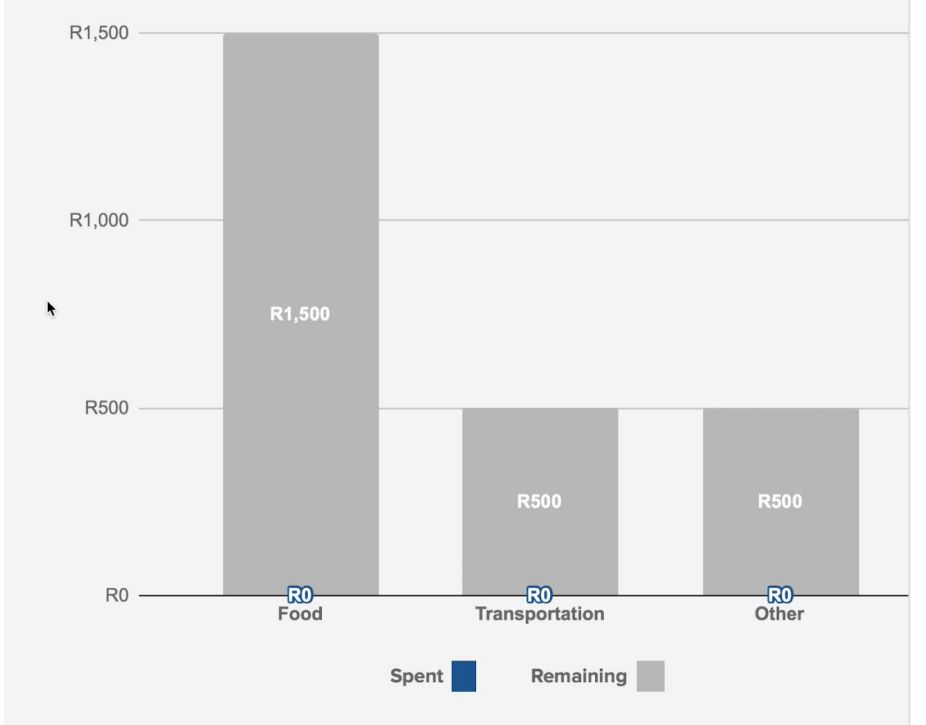
4	Planned spend	R2,500	R1,500
---	---------------	--------	--------

5	Actual spending	R0
---	-----------------	----

[illegible]

10		Budget	Spent	Remaining	Budget used
11	Totals	R2,500	R0	R2,500	
12					
13	Food	R1,500	R0	R1,500	0%
14	Transportation	R500	R0	R500	0%
15	Other	R500	R0	R500	0%

Spending



Simulator ▾

402167xxxxx8016

Back

Cents amount



50000

Currency



South African Rand


Merchant
category

Grocery Stores, Supermarkets



Merchant name

Spar

Merchant city

Cape Town

Run in Sandbox

Console

Before transaction

After transaction

Script is: Active | Published

Deactivate

Publish

```

1  const beforeTransaction = async (authorization) => { };
2
3  const afterTransaction = async (transaction) => {
4      const sheetUrl = 'https://script.google.com/a/offerzen.com/mac
5      const payload = {
6          row: [
7              moment(transaction.dateTime).format('YYYY-MM-DD'),
8              moment(transaction.dateTime).format('HH:mm'),
9              helpers.format.decimal(transaction.centsAmount / 100,
10             transaction.currencyCode,
11             transaction.merchant.category.name,
12             transaction.merchant.name,
13             transaction.merchant.city
14         ]
15     };
16
17     const response = await fetch(sheetUrl, {
18         method: 'POST',
19         headers: { 'Content-Type': 'application/json' },
20         body: JSON.stringify(payload),
21     });
22     return true;
23 };
24

```



	Food										
	A	B	C	D	E	F	G	H	I	J	K
1	Category	Date	Amount	Merchant	Location						
2	Food	2019-11-28	R500.00	Spar	Cape Town						
3	Other	2019-11-28	R200.00	Spar	Cape Town						
4	Food	2019-11-28	R200.00	Spar	Cape Town						
5	Food	2019-11-28	R200.00	Uber Eats	Cape Town						
6	Food	2019-11-28	R200.00	Uber Eats	Cape Town						
7	Other	2019-11-28	R100.00	Uber Eats	Cape Town						
8	Food	2019-11-28	R500.00	Uber Eats	Cape Town						
9	Food	2019-11-28	R500.00	Spar	Cape Town						
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											

1

2

4

5

9

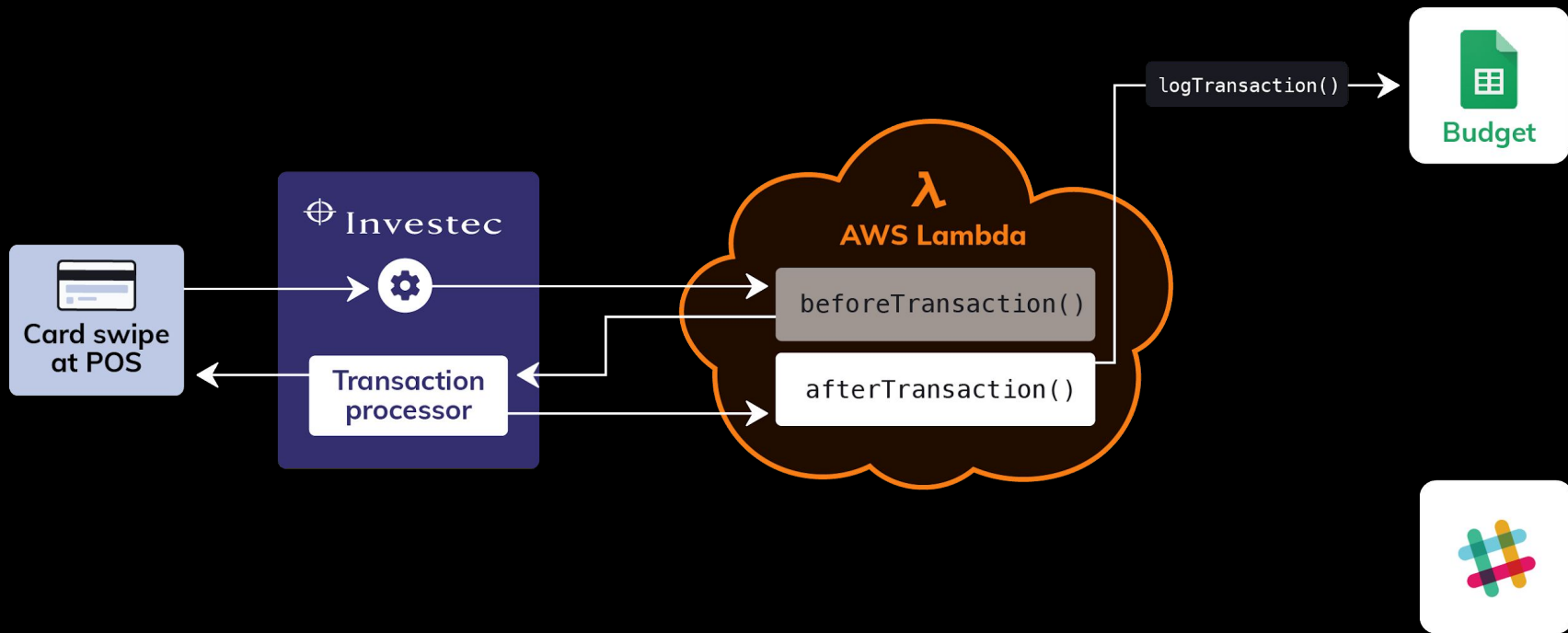
10



2. I want to get notified of my
remaining limit

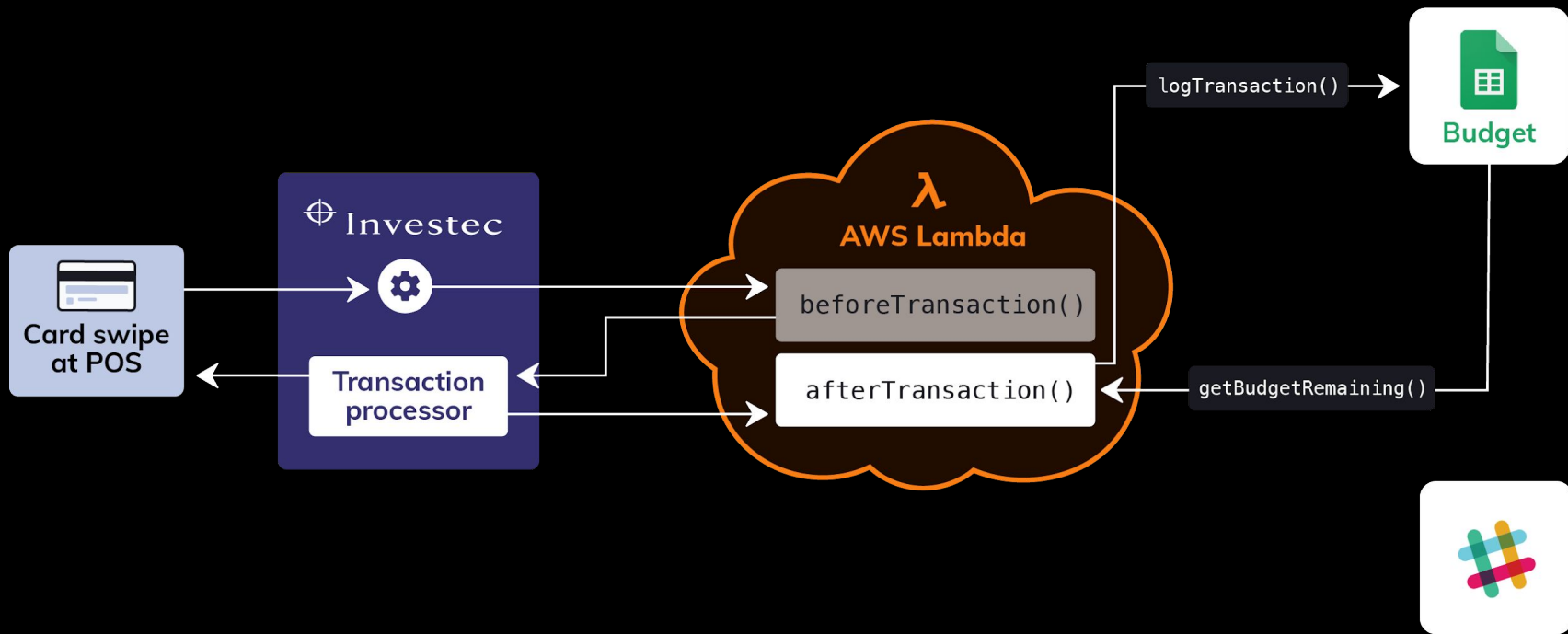
Reading data from sheets and posting to Slack in

`afterTransaction()`



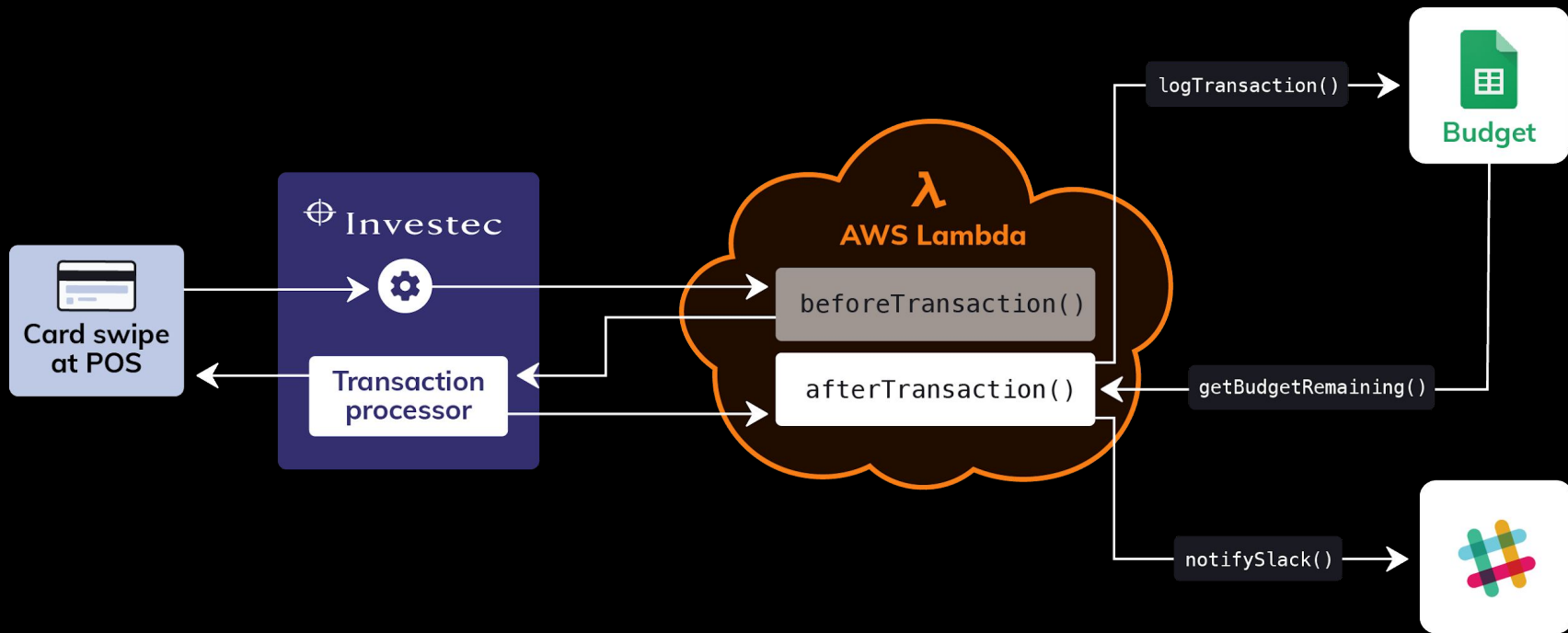
Reading data from sheets and posting to Slack in

`afterTransaction()`



Reading data from sheets and posting to Slack in

`afterTransaction()`



Simulator ▾

402167xxxxx8016

Back

Cents amount



20000

Currency



South African Rand

Merchant
category

Fast Food Restaurants



Merchant name

Uber Eats

Merchant city

Cape Town

Run in Sandbox

Console

Before transaction

After transaction

Script is: Active | Published

Deactivate

Publish

```
35     default:
36         return "Other";
37     }
38 }
39
40 const getRemainingBudget = async (budgetCategory) => {
41     let remainingBudgetCell = {
42         "Food": "E13",
43         "Transportation": "E14",
44         "Other": "E15"
45     }[budgetCategory]
46
47     const query = `?sheet=My+Budget&range=${remainingBudgetCell}`;
48
49     const response = await fetch(sheetUrl + query, {
50         method: 'GET',
51         headers: { 'Content-Type': 'application/json' }
52     });
53     const json = await response.json();
54
55     return parseInt(json.data[0][0])
56 }
57
58 const postToSlack = async (message) => { ...
59 }
60
61 const afterTransaction = async (transaction) => {
62     await postToSlack(transaction)
63 }
```



Simulator ▾

402167xxxxx8016

Back

Cents amount



20000

Currency



South African Rand

Merchant
category

Fast Food Restaurants



Merchant name

Uber Eats

Merchant city

Cape Town

Run in Sandbox

Console

Before transaction

After transaction

Script is: **Active** | Published

Deactivate

Publish

```
48
49   const response = await fetch(sheetUrl + query, {
50     method: 'GET',
51     headers: { 'Content-Type': 'application/json' }
52   });
53   const json = await response.json();
54
55   return parseInt(json.data[0][0])
56 }
57
58 const postToSlack = async (message) => {
59   const slackUrl = "https://hooks.slack.com/services/T0DC47CQM/B
60   const response = await fetch(slackUrl, {
61     method: 'POST',
62     headers: { 'Content-Type': 'application/json' },
63     body: JSON.stringify({ channel: "", text: message }),
64   });
65 }
66
67 const afterTransaction = async (transaction) => {
68   await logTransaction(transaction);
69
70   const budgetCategory = getBudgetCategory(transaction);
71   const remainingBudget = await getRemainingBudget(budgetCategor
72
73   await postToSlack(`You have R${remainingBudget} left for ${bud
74
75   return true;
```





Offerzen

dan

You have R400 left for Food this week

Reply

Weekly Budget

2

Summary

Planned spend R2,500

Actual spending R1,300

6

7

8

Spending categories

	Budget	Spent	Remaining	Budget used
Totals	R2,500	R1,300	R1,200	
Food	R1,500	R1,100	R400	73%
Transportation	R500	R0	R500	0%
Other	R500	R200	R300	40%

16

17

18

19

20

21

22

23

24

25

26

27

28

Spending

R1,500

R400

R1,000

R1,100

R500

R0

R500

R300

R200

Food

Transportation

Other

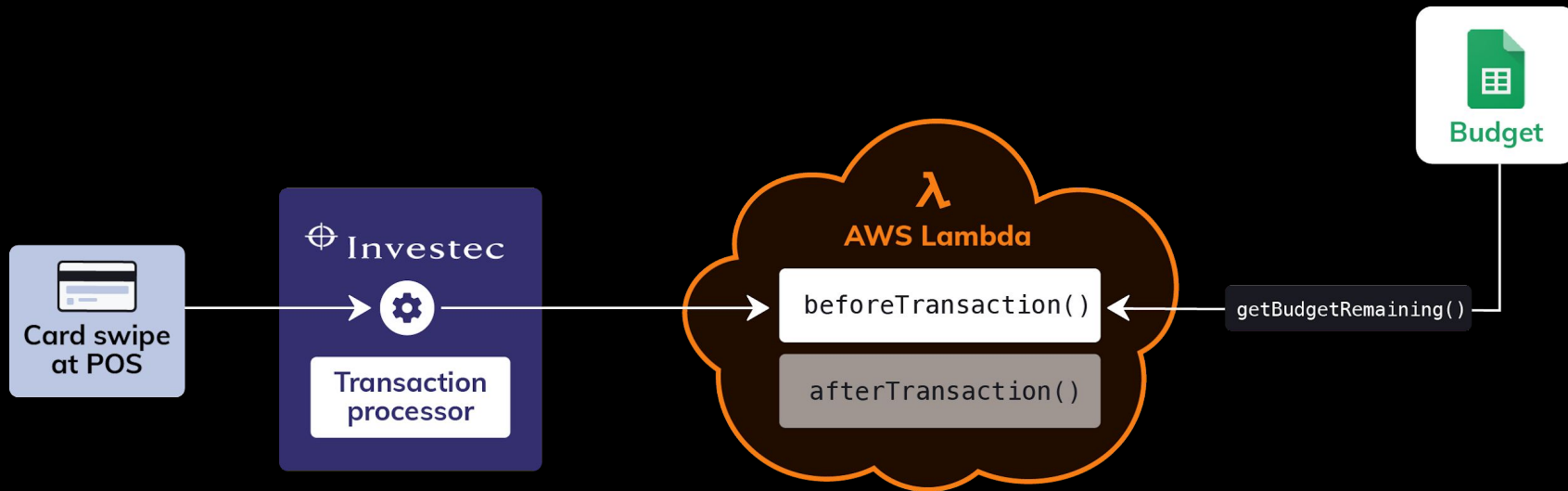
Spent

Remaining

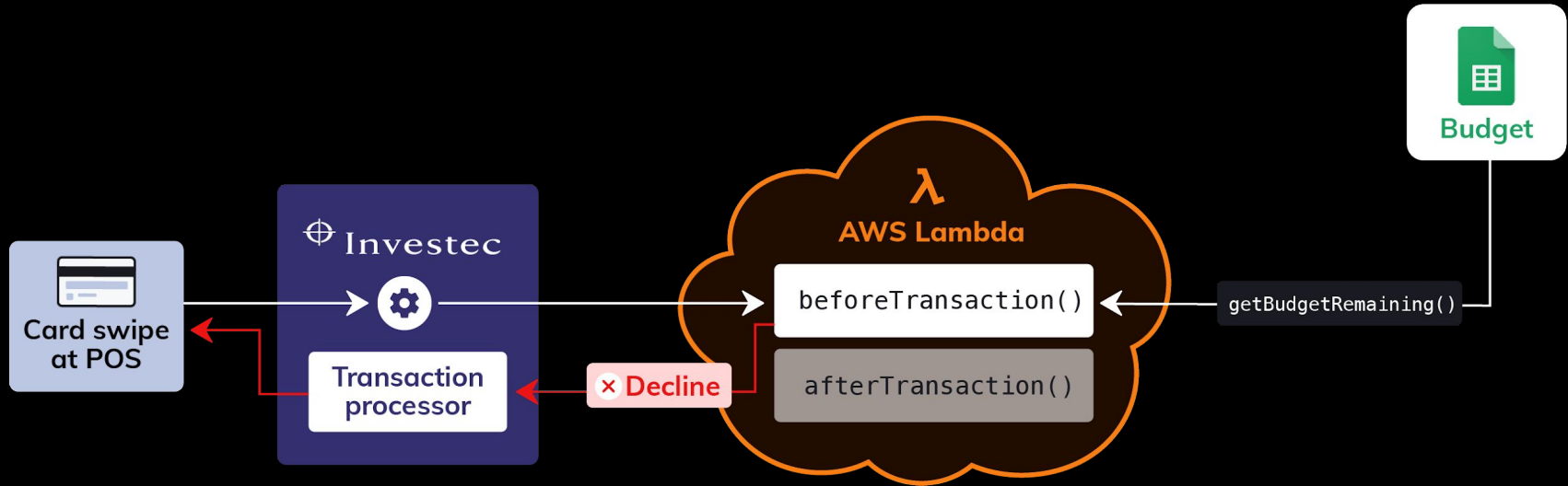


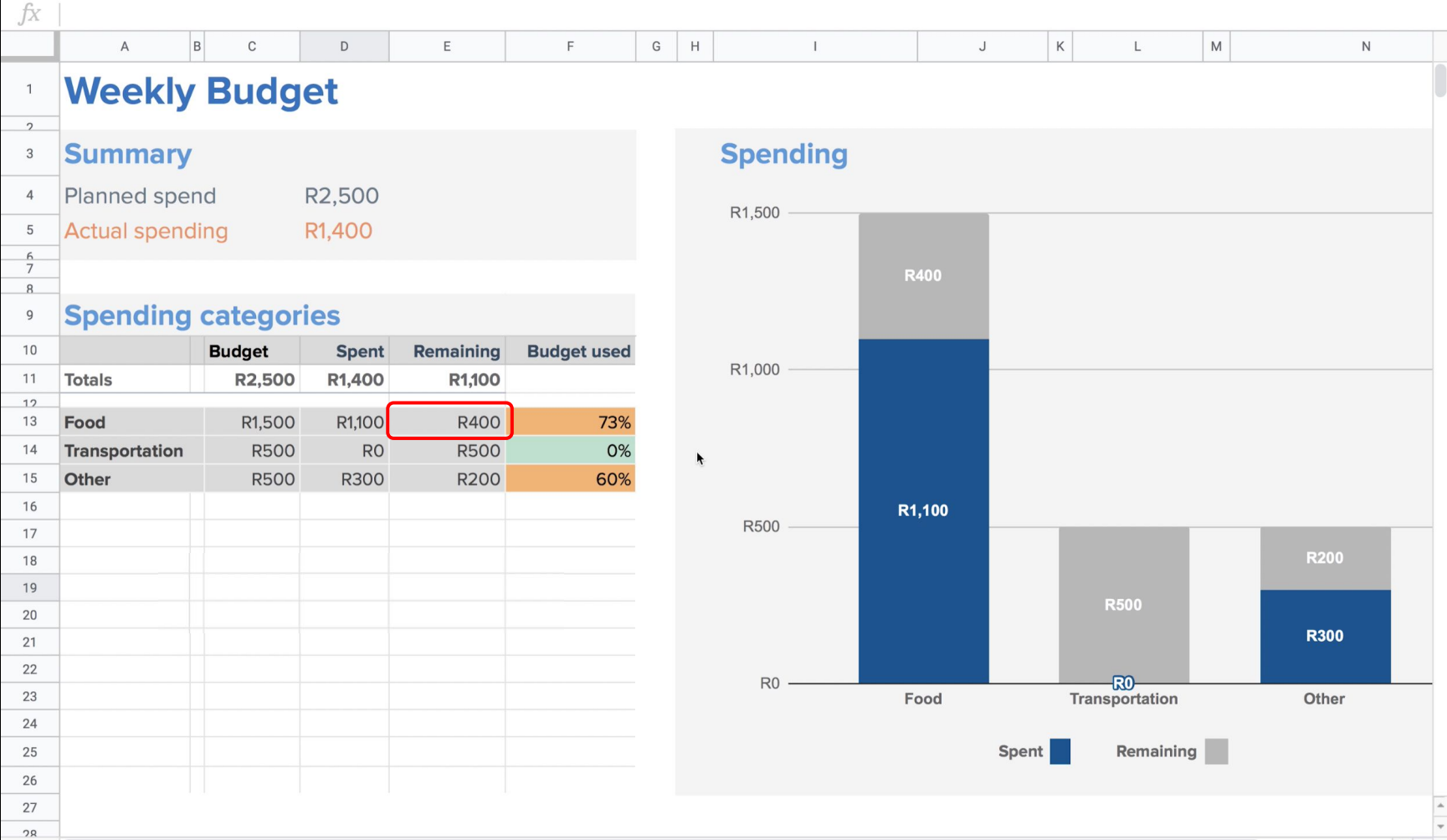
3. I want to stop myself from
going over budget

Using conditional logic in `beforeTransaction()`



Using conditional logic in `beforeTransaction()`





Simulator ▾

402167xxxxx8016

Back

Cents amount



50000

Currency



South African Rand

Merchant
category

Fast Food Restaurants



Merchant name

Uber Eats

Merchant city

Cape Town

Run in Sandbox

Console

Before transaction

After transaction

Script is: **Active** | Published

Deactivate

Publish

```
63 }
64
65
66 const beforeTransaction = async (authorization) => {
67   const budgetCategory = getBudgetCategory(authorization);
68   const remainingBudget = await getRemainingBudget(budgetCategory);
69   I
70   const authorized = authorization.centsAmount < (remainingBudget);
71
72   if(!authorized) {
73     await postToSlack(`You are over budget for ${budgetCategory}`);
74   }
75
76   return authorized;
77 };
78
79 + const afterTransaction = async (transaction) => { ...
90 };
91
92
```



category

Merchant name

Uber Eats

Merchant city

Cape Town

Run in Sandbox

Console

Before transaction

After transaction

```
{
  "sandbox": true,
  "type": "before_transaction",
  "authorizationApproved": false,
  "logs": [],
  "smsCount": 0,
  "emailCount": 0,
  "pushNotificationCount": 0,
  "createdAt": "/Date(1574938836813)/",
  "startedAt": "/Date(1574938836812)/",
  "completedAt": "/Date(1574938839313)/",
  "updatedAt": "/Date(1574938836813)/"
}
```

Script is: Active | Published

Deactivate

Publish

```
67   const budgetCategory = getBudgetCategory(authorization);
68   const remainingBudget = await getRemainingBudget(budgetCategory);
69
70   const authorized = authorization.centsAmount < (remainingBudget);
71
72   if(!authorized) {
73     await postToSlack(`You are over budget for ${budgetCategory}`);
74   }
75
76   return authorized;
77 };
78
79 + const afterTransaction = async (transaction) => { ...
90 };
91
92
```

1

2

4

5

9

10



Offerzen
dan
You have R900 left for Food this week

[Reply](#)

Currency



South African Rand

Merchant
category

Fast Food Restaurants



Merchant name

Uber Eats

Merchant city

Cape Town

[Run in Sandbox](#)[Console](#)[Before transaction](#)[After transaction](#)

```
{
  "sandbox": true,
  "type": "before_transaction",
  "authorizationApproved": true,
  "logs": [],
  "smsCount": 0,
  "emailCount": 0,
  "pushNotificationCount": 0
}
```

Script is: **Active** | Published[Deactivate](#)[Publish](#)

```
67   const budgetCategory = getBudgetCategory(authorization);
68   const remainingBudget = await getRemainingBudget(budgetCategory);
69
70   const authorized = authorization.centsAmount < (remainingBudget);
71
72   if(!authorized) {
73     await postToSlack(`You are over budget for ${budgetCategory}`);
74   }
75
76   return authorized;
77 };
78
79 + const afterTransaction = async (transaction) => {
80 };
81
82
```



All the things



Expensify



Office 365



amazon alexa



FOREX.com



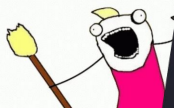
aws



Google Assistant



salesforce



Investec

```
// This function runs during the card transaction authorization
flow.
// It has a limited execution time, so keep any code short-running.
const beforeTransaction = async (authorization) => {
  const randAmount = authorization.centsAmount / 100;
  if (authorization.merchant.category.key === 'Travel Agencies,
    Tour Operators' || randAmount < 500) {
    return true;
  } else {
    const sms = await notifications.sms({
      mobile: '+27831234567',
      body: 'Fraudulent transaction'
    })
    sms.send();
    return false; // Decline transaction
  }
};

// This function runs after a transaction was successful.
const afterTransaction = async (transaction) => {
  const amount = formatCurrency(authorization.centsAmount,
    authorization.currencyCode);
  const message = amount + ' spent at ' + authorization.merchant.name;
  notifications.sendSMS(message, '+27821234567');
  notifications.sendEmail(message, 'example@example.com');
  notifications.sendPush(message);
};
```



Community Beta

offerzen.com/community/investec

Questions?
