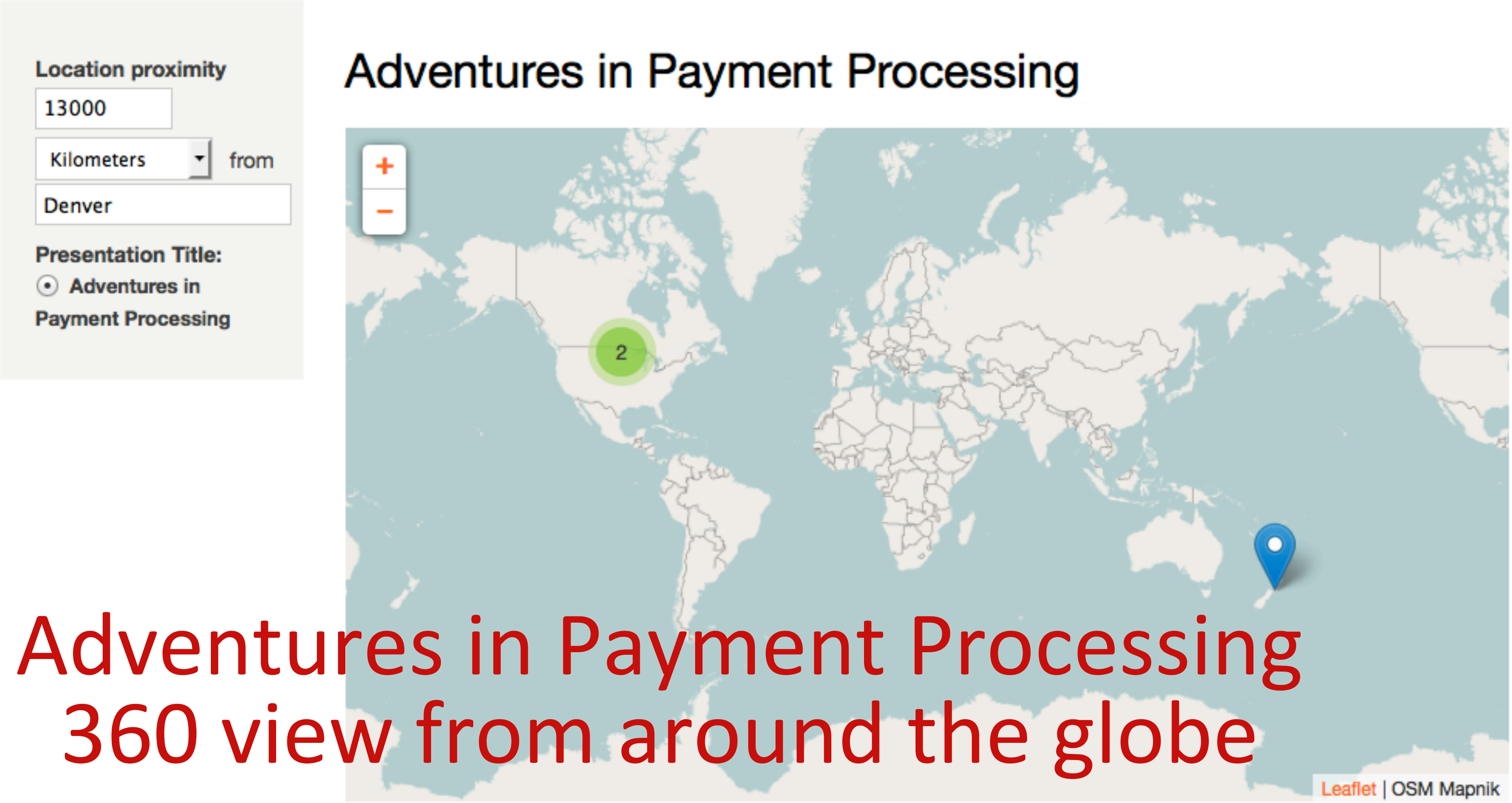


Location proximity

13000

Kilometers from

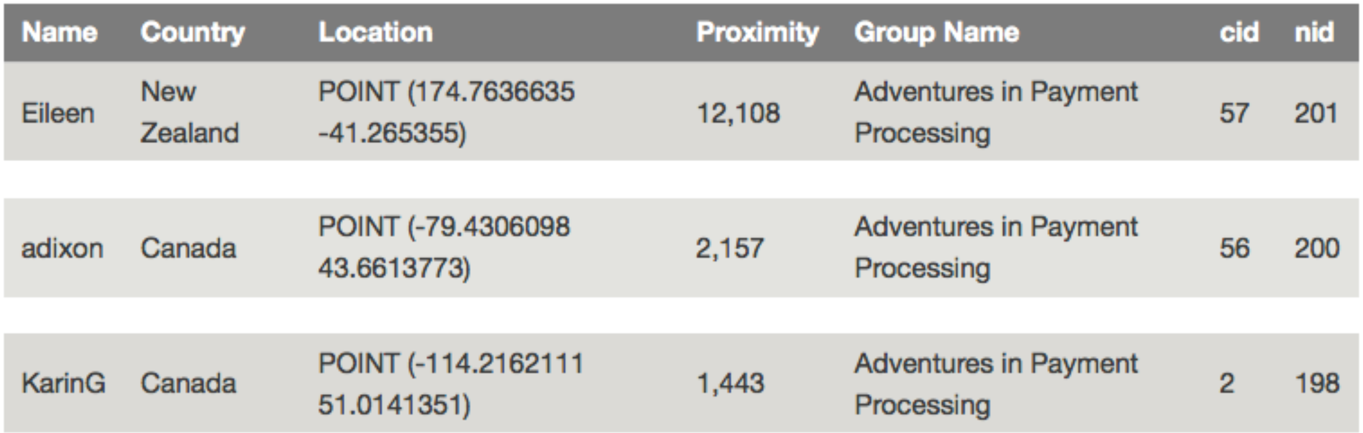
Denver

Presentation Title:

● Adventures in Payment Processing

Adventures in Payment Processing

Adventures in Payment Processing
360 view from around the globe



Name	Country	Location	Proximity	Group Name	cid	nid
Eileen	New Zealand	POINT (174.7636635 -41.265355)	12,108	Adventures in Payment Processing	57	201
adixon	Canada	POINT (-79.4306098 43.6613773)	2,157	Adventures in Payment Processing	56	200
KarinG	Canada	POINT (-114.2162111 51.0141351)	1,443	Adventures in Payment Processing	2	198

Goals for Today

- History: Payment Processing in CiviCRM
- Current state of affairs – based on true stories!
- Moving onward and forward

There will be no time for any questions…

Payment Processing

- Is there anything more important? … to so many organizations than the ability to process a variety of live payments across all CiviCRM pathways?

In the beginning...

There was PayPal…

- PayPal was written directly into Core AS IF it was the only Payment Processor out there!
- and – fair enough - it essentially was!

civispace-0.8.3

- was certainly meant to be processor agnostic
- over time a lot of PayPal assumptions went into Core and not everything got refactored into PayPal specific bits

PayPal Became Entangled in Core

```
$ grep -inr paypal civicrm | wc –l
```

441

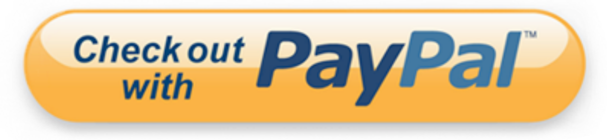
Example:

CRM/Event/Form/Participant.php:277:

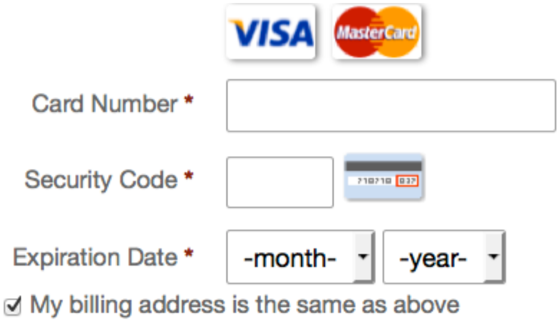
```
if ($paymentProcessor['payment_processor_type'] == 'PayPal' && !
    $paymentProcessor['user_name']) {
```

3 Flavors:

PayPal Express



PayPal Pro



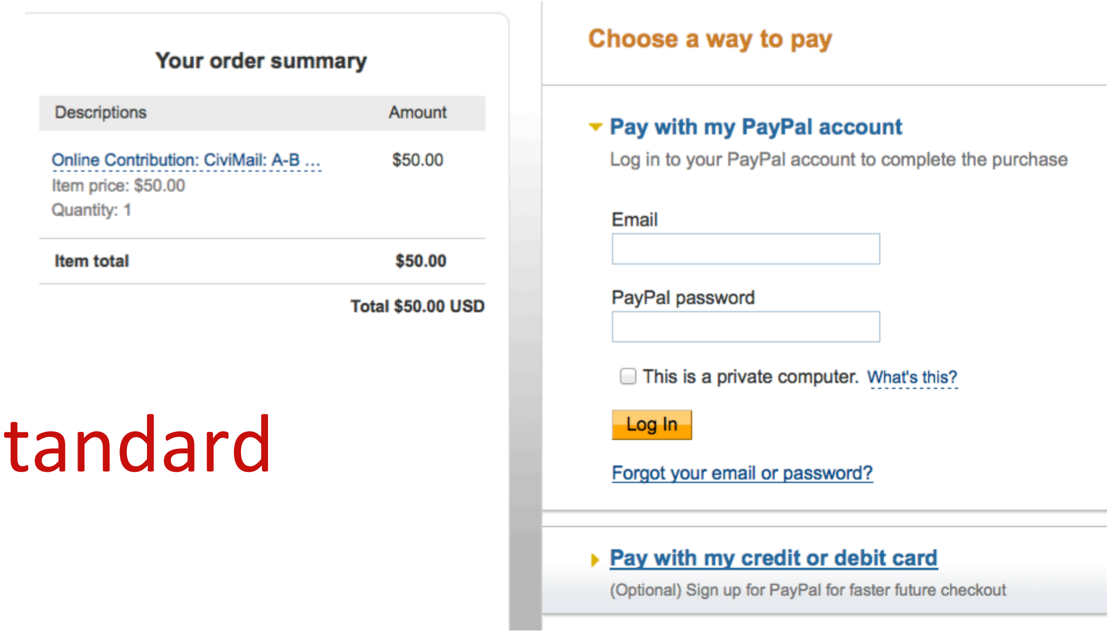
Card Number *

Security Code *

Expiration Date * -month- -year-

☑ My billing address is the same as above

PayPal Standard



Your order summary

Descriptions	Amount
Online Contribution: CiviMail: A-B ... Item price: $50.00 Quantity: 1	$50.00
Item total	**$50.00**

Total $50.00 USD

Choose a way to pay

Pay with my PayPal account

Log in to your PayPal account to complete the purchase

Email

PayPal password

□ This is a private computer. What's this?

Log In

Forgot your email or password?

Pay with my credit or debit card

(Optional) Sign up for PayPal for faster future checkout

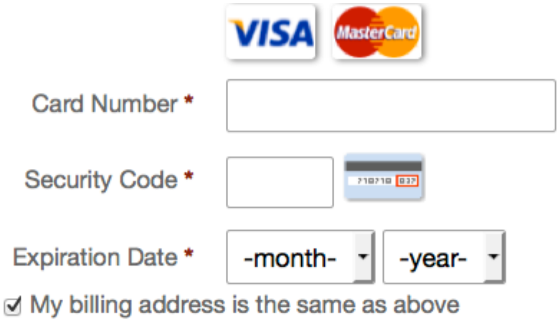
3 Flavors:

↓

3 Buckets

PayPal Express

→ express



Card Number *

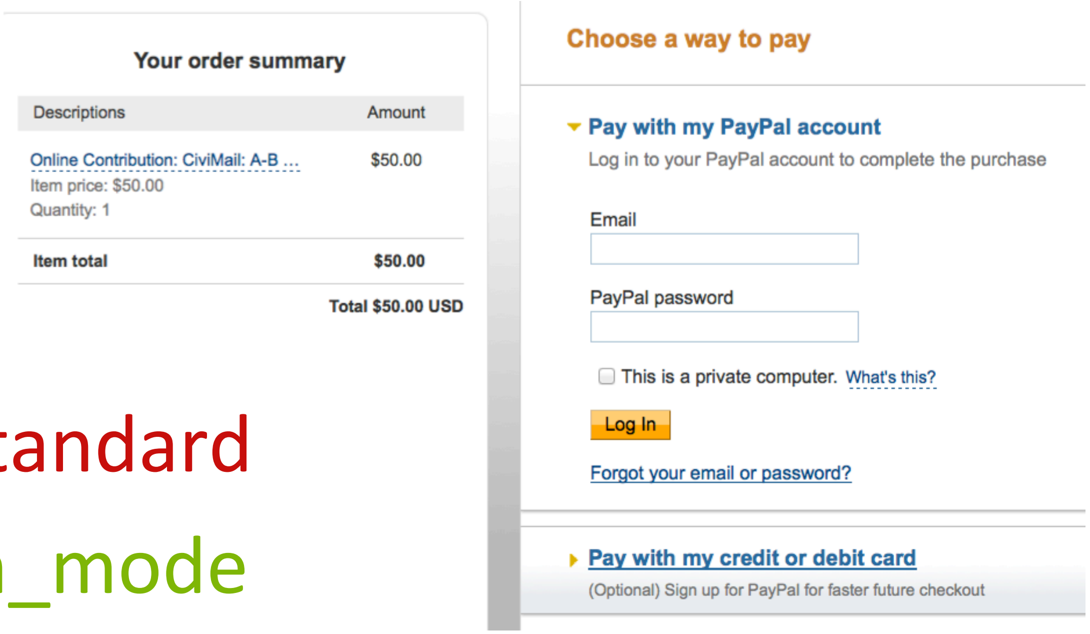
Security Code *

Expiration Date * -month- -year-

☑ My billing address is the same as above

PayPal Pro

→ billing_mode



Your order summary

Descriptions	Amount
Online Contribution: CiviMail: A-B ... Item price: $50.00 Quantity: 1	$50.00
Item total	**$50.00**

Total $50.00 USD

Choose a way to pay

Pay with my PayPal account

Log in to your PayPal account to complete the purchase

Email

PayPal password

□ This is a private computer. What's this?

Log In

Forgot your email or password?

Pay with my credit or debit card

(Optional) Sign up for PayPal for faster future checkout

PayPal Standard

→ form_mode

3 Flavors:

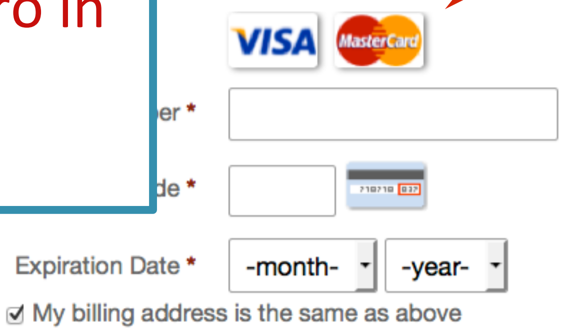
→

3 Buckets

PayP

NOT an omission
Cannot do AMEX with PayPal Pro in Canada

PayP



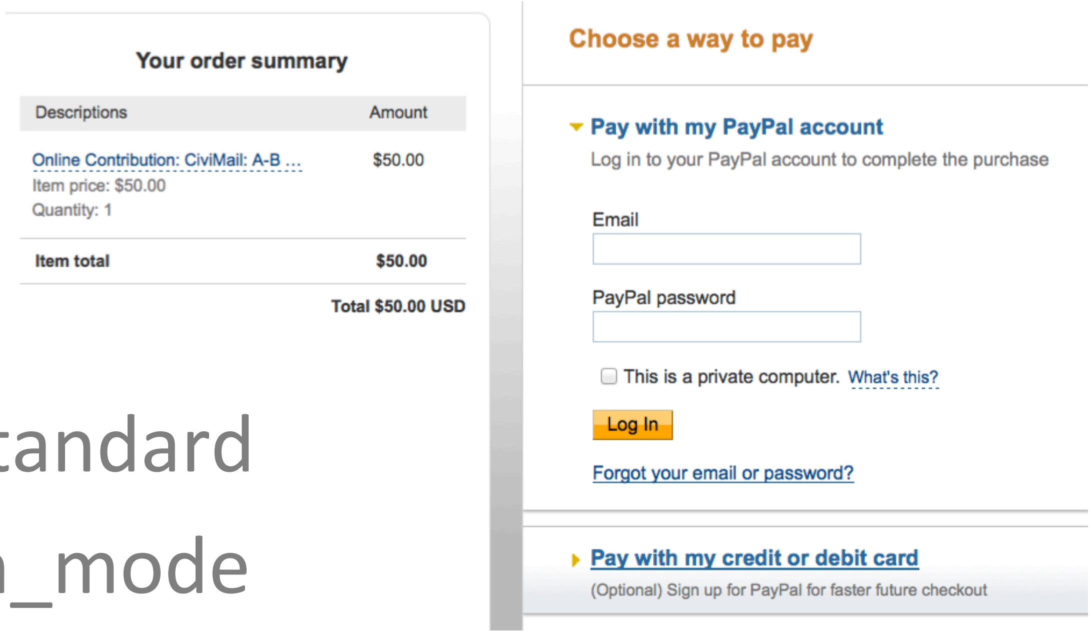
er *

de *

Expiration Date * -month- -year-

☑ My billing address is the same as above

→ billing_mode



Your order summary

Descriptions	Amount
Online Contribution: CiviMail: A-B ... Item price: $50.00 Quantity: 1	$50.00
Item total	**$50.00**

Total $50.00 USD

Choose a way to pay

▾ Pay with my PayPal account

Log in to your PayPal account to complete the purchase

Email

PayPal password

□ This is a private computer. What's this?

Log In

Forgot your email or password?

▸ Pay with my credit or debit card

(Optional) Sign up for PayPal for faster future checkout

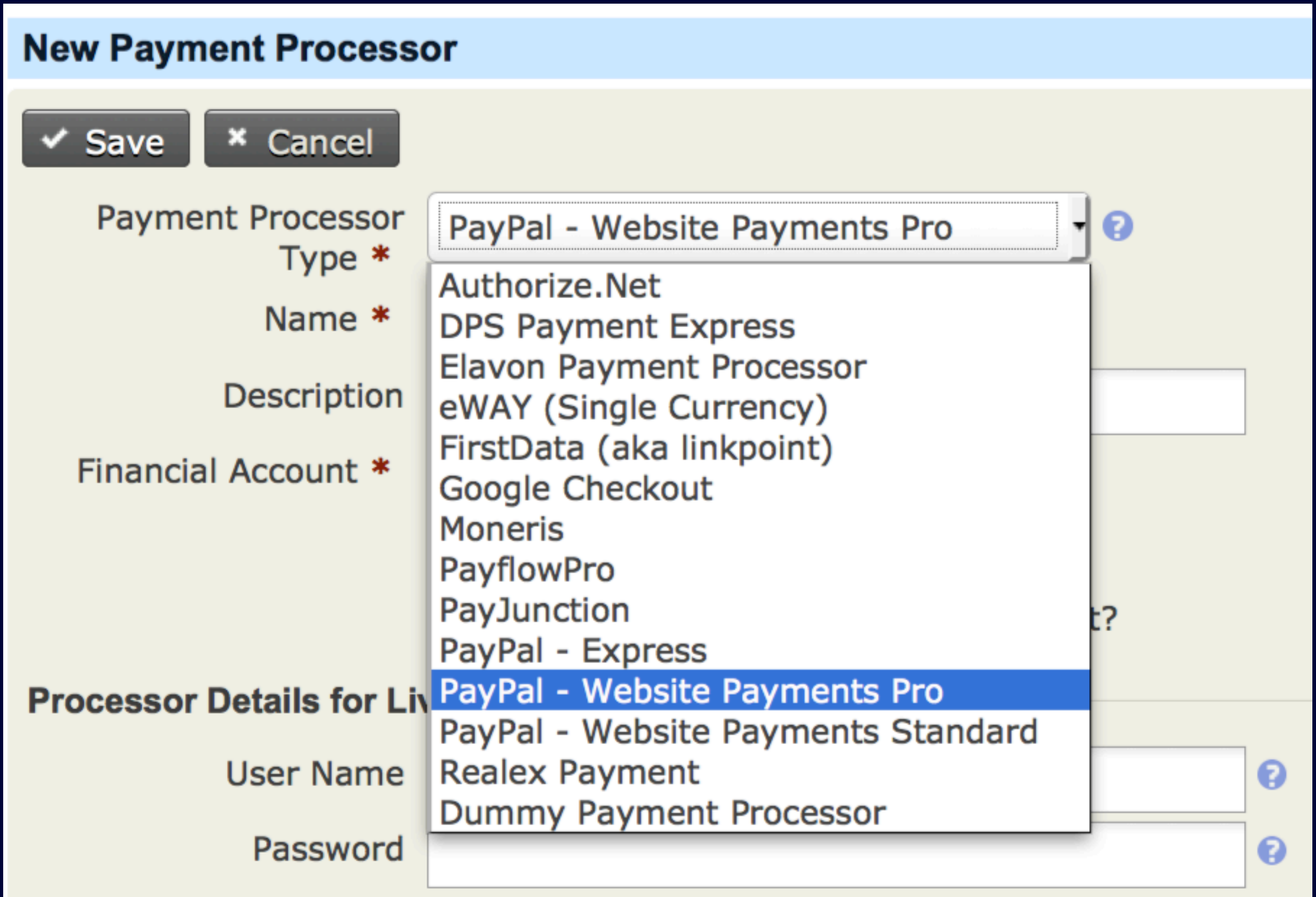
PayPal Standard

→ form_mode

Introduction of PayPal Vagaries

- PayPal standard does not support backoffice – so offsite (hosted) payments via backoffice do not work, period.
- PayPal has delayed payment for recurring contributions – so assume delayed payment for all recurring contributions
- PayPal does IPNs – so there is no place in Core to store/handle tokens (customer codes)

Processors Added to Core



New Payment Processor

Payment Processor Type * PayPal - Website Payments Pro

Name *

Description

Financial Account *

Processor Details for Li

User Name

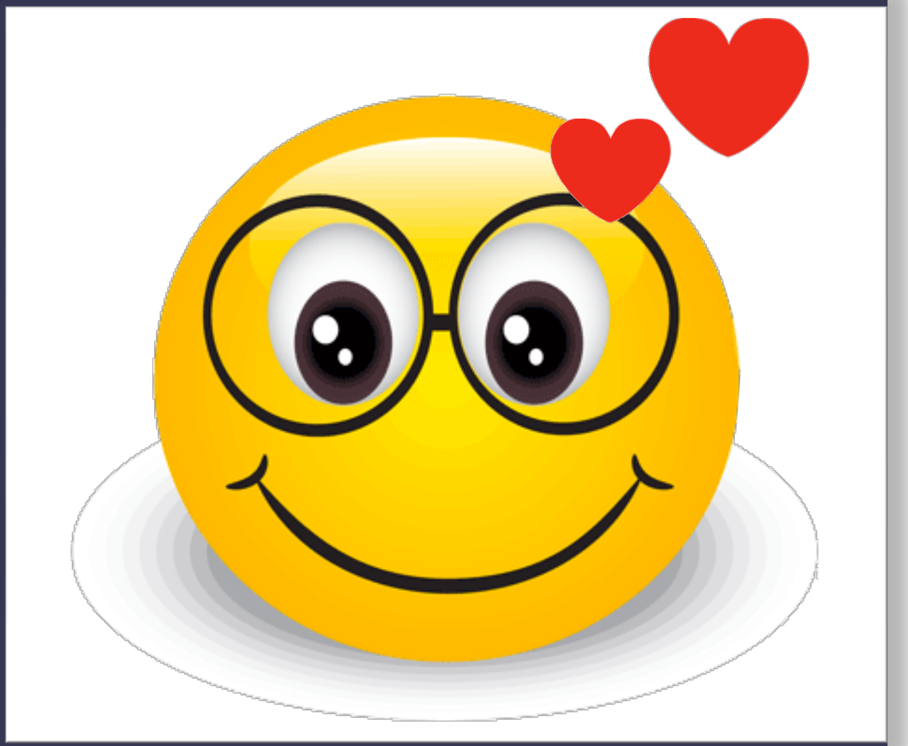
Password

BaseIPN

- Horrible things happened
  kept on happening
  to the BaseIPN

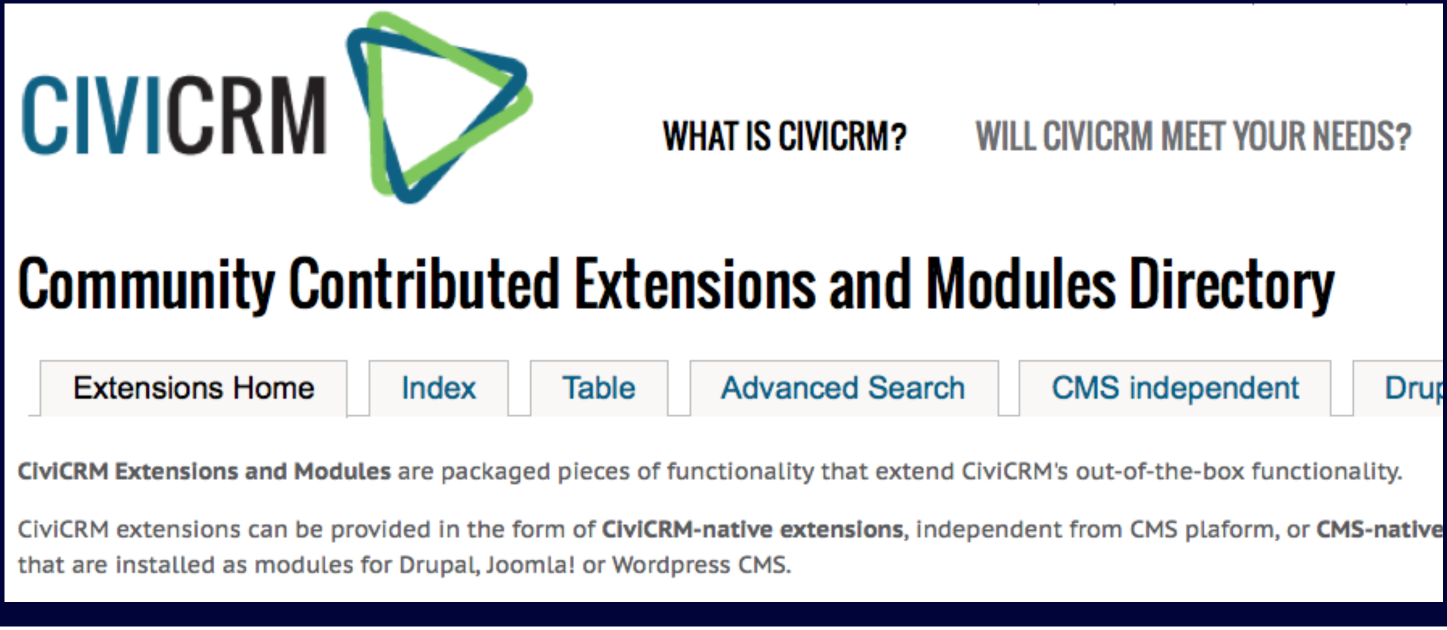
Extensions Framework Introduced

- Developers LOVE extensions!
- Organizations LOVE to choose/work with Payment Processor of their choice!
- Payment Processor Extensions – first off the blocks!



and Kept Evolving

- And over the years Payment Processor extensions started trying to do more complex things: e.g. ACH/EFT (direct debit), encrypted swipe, direct post, backoffice recurring, etc.



Community Contributed Extensions and Modules Directory

CiviCRM Extensions and Modules are packaged pieces of functionality that extend CiviCRM's out-of-the-box functionality.

CiviCRM extensions can be provided in the form of **CiviCRM-native extensions**, independent from CMS plaform, or **CMS-native** that are installed as modules for Drupal, Joomla! or Wordpress CMS.

Current State of Affairs

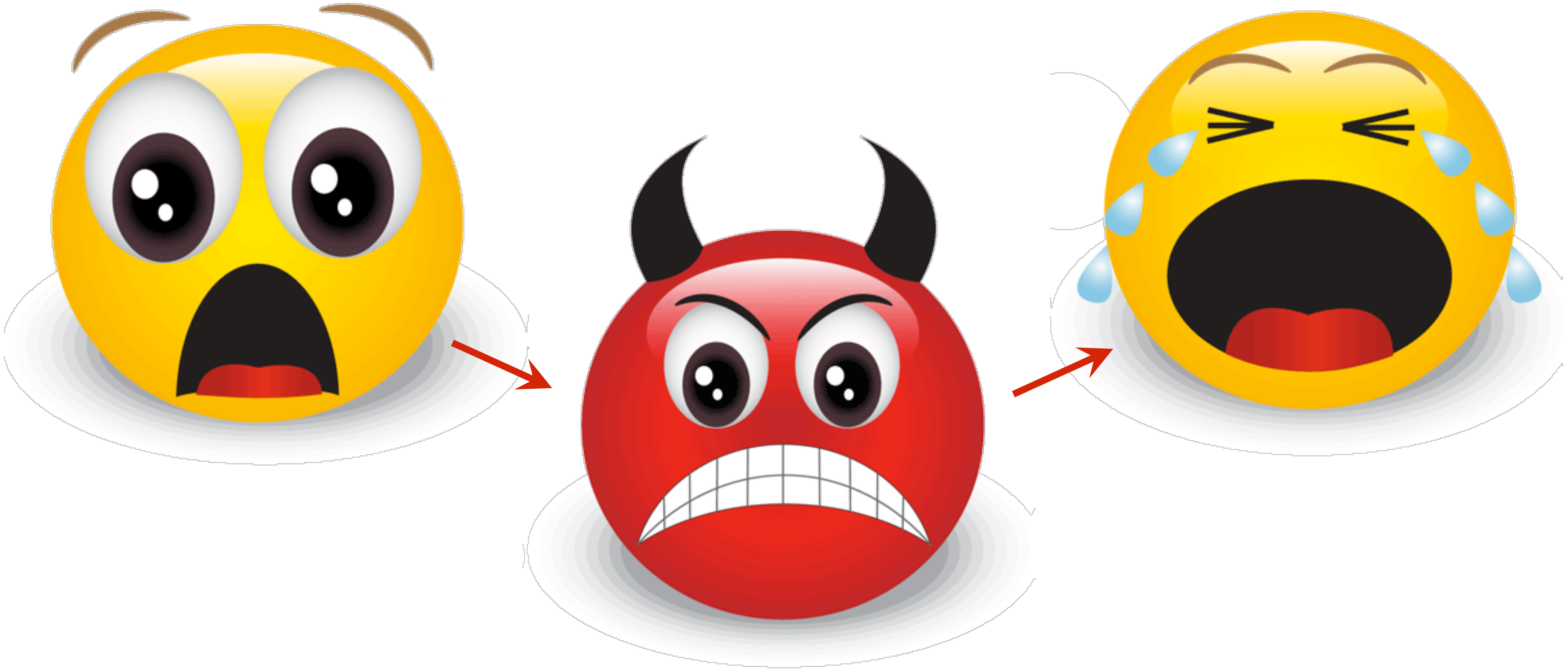
- What sort of Payment Processors are there now – and what do they look like compared to CiviCRM assumptions?

Type	Examples (not a complete listing)	Fit with built-in assumptions
Simple onsite Credit Card	PayPal Pro, Eway Standard	✓
Simple redirect	DPS Payment Express	✓
Random PayPal specific	PayPal Express (only)	✓

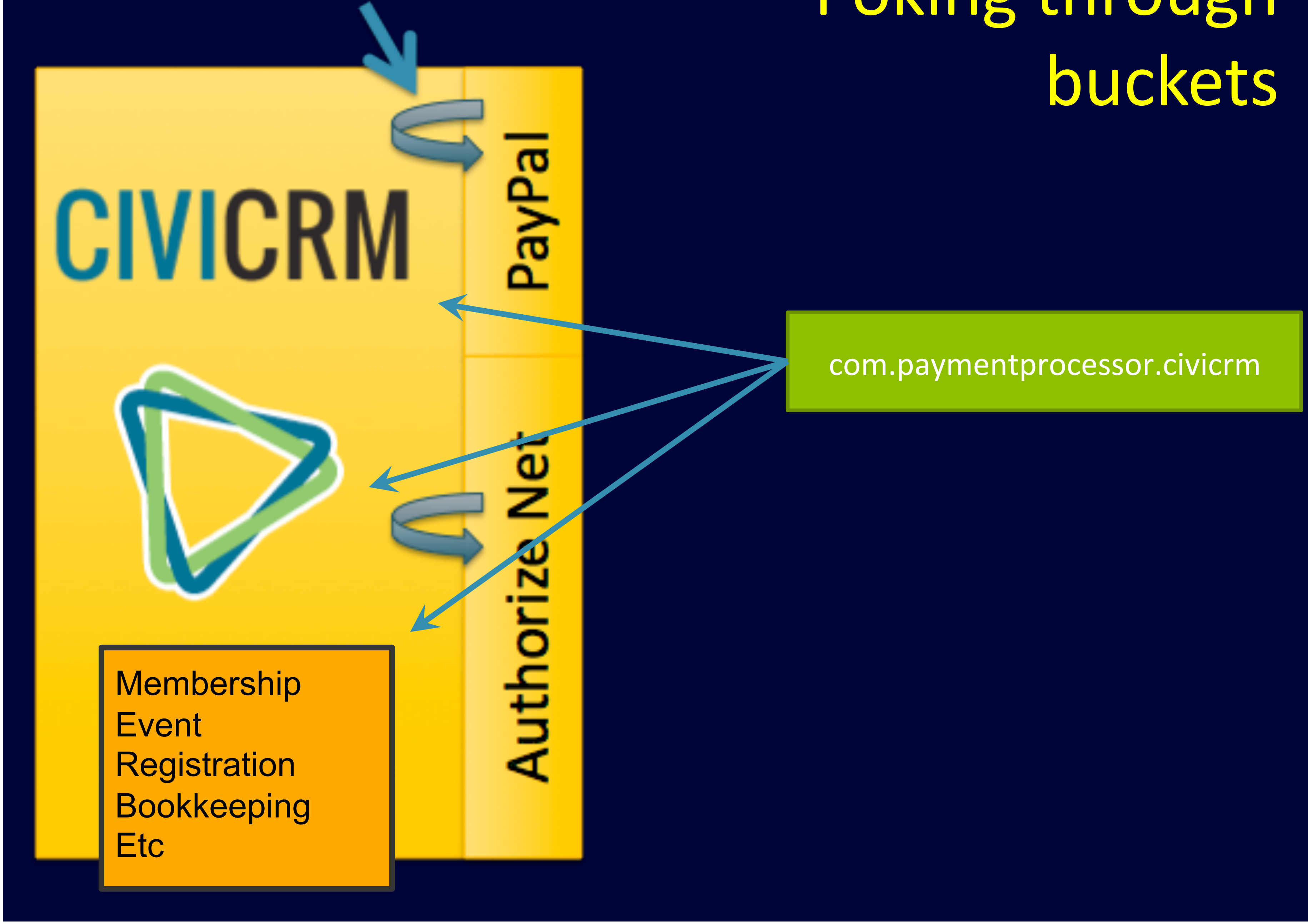
ACH/EFT (direct debit)	iATS Payments, Moneris (last)	✗
JS intercept	Eway client side (3) encryption, (Stripe)	✗
Transparent redirect	Cybersource (2)	✗
Off-site post	Paybox (1)	✗
Payment tokenization	Beanstream, iATS Payments, eWay recurring	✗
Fly-ins	Global Collect (4)	✗

CiviCRM **thinks** It can build a form based on assumptions about the processor but the assumptions are based on PayPal

CiviCRM **thinks** It can build a form based on assumptions about the processor but the assumptions are based on PayPal

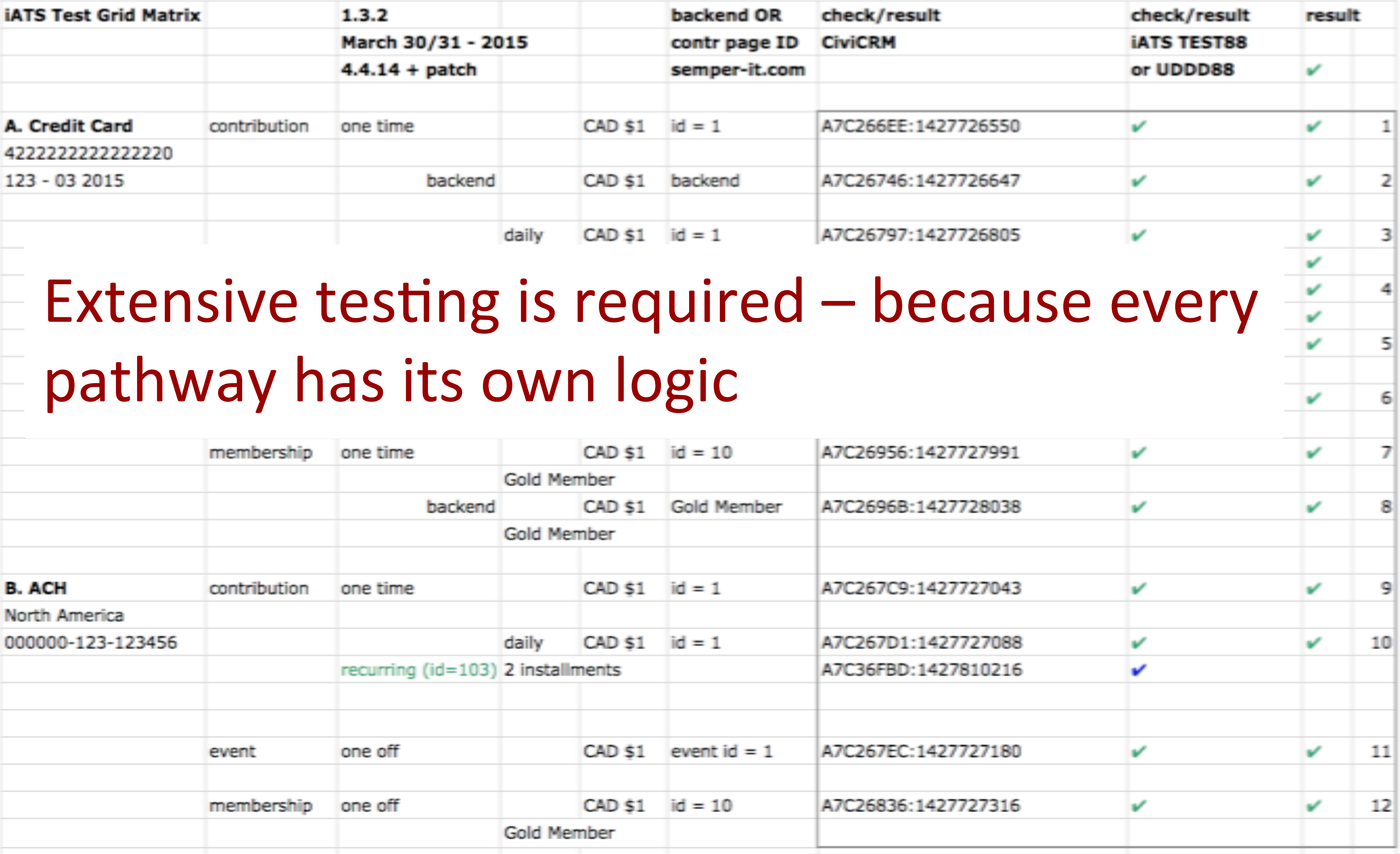


Poking through buckets



Membership
Event
Registration
Bookkeeping
Etc

iATS Extension – Test Matrix - p1



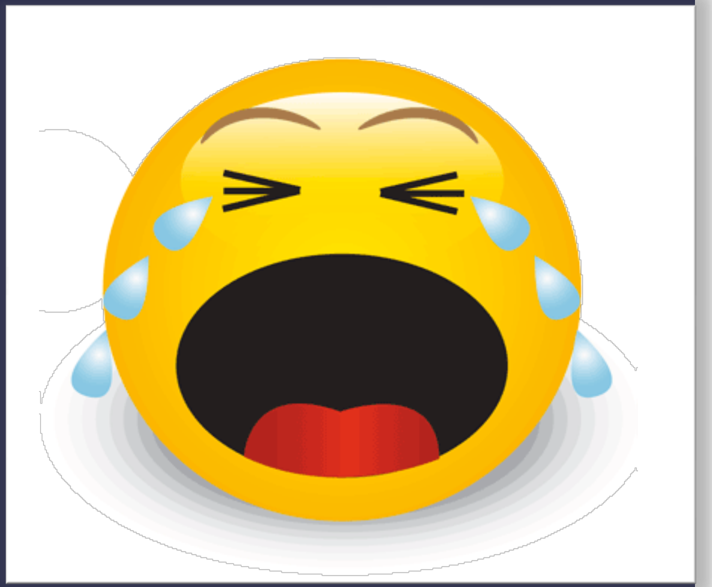
iATS Test Grid Matrix		**1.3.2**			**backend OR**	**check/result**	**check/result**	**result**	
		March 30/31 - 2015			**contr page ID**	**CiviCRM**	**iATS TEST88**		
		4.4.14 + patch			**semper-it.com**		**or UDDD88**	✓	
A. Credit Card	contribution	one time		CAD $1	id = 1	A7C266EE:1427726550	✓	✓	1
4222222222222220									
123 - 03 2015		backend		CAD $1	backend	A7C26746:1427726647	✓	✓	2
			daily	CAD $1	id = 1	A7C26797:1427726805	✓	✓	3
								✓	
								✓	4
								✓	
								✓	5
								✓	6
	membership	one time		CAD $1	id = 10	A7C26956:1427727991	✓	✓	7
			Gold Member						
		backend		CAD $1	Gold Member	A7C2696B:1427728038	✓	✓	8
			Gold Member						
B. ACH	contribution	one time		CAD $1	id = 1	A7C267C9:1427727043	✓	✓	9
North America									
000000-123-123456			daily	CAD $1	id = 1	A7C267D1:1427727088	✓	✓	10
		recurring (id=103)	2 installments			A7C36FBD:1427810216	✓		
	event	one off		CAD $1	event id = 1	A7C267EC:1427727180	✓	✓	11
	membership	one off		CAD $1	id = 10	A7C26836:1427727316	✓	✓	12
			Gold Member						

Extensive testing is required – because every pathway has its own logic

But… organizations want to:

- Membership +
- Recurring +
- ACH/EFT (Direct Debit) +
- by admin staff in backoffice!

Sorry…



IF

$paymentParams[contribution_status_id] = 1

All subsequent logic should reside in CiviCRM core and should be Payment Processor agnostic

- event registration (online or backoffice, credit card or direct debit) - ✔
- membership (online or backoffice or recurring online, or recurring backoffice, credit card or direct debit) - ✔

Onward and Forward: Recently

- 4.6 code changes: decoupling of payment processor type from assumptions: power to the processors – does this processor support back office transactions? Let's ask it!

```
if ($processor->supports('BackOffice')) {
    // sing & dance!
}
```

Override-able in 4.6

- Payment Processor Extensions can now override these assumptions:

```
$processor->supportsLiveMode()
```

Override-able in 4.6

- Payment Processor Extensions can now override these assumptions:

```
$processor->supportsFutureRecurStartDate()
```

Override-able in 4.6

- Payment Processor Extensions can now override these assumptions:

```
$processor->supports('BackOffice')
```

Override-able in 4.6

- Payment Processor Extensions can now override these assumptions:

```
$processor->getPaymentFormFields()
```

Override-able in 4.6

- Payment Processor Extensions can now override these assumptions:

```
$processor->getPaymentFormFieldsMetadata()
```

Override-able in 4.6

- Payment Processor Extensions can now override these assumptions:

$processor->getPaymentTypeLabel() (e.g Credit Card)

Override-able in 4.6

- Payment Processor Extensions can now override these assumptions:

```
$processor->getPaymentTypeName() (e.g credit_card)
```

Override-able in 4.6

- Payment Processor Extensions can now override these assumptions:

```
$processor->validatePaymentInstrument($values,
   $errors);
```

Death to Singletons!

Old Way

-
  ```
  $paymentProcessor =
  CRM_Financial_BAO_PaymentProcessor::
          getPayment($payment_processor_id,
                         'live');
  ```

-
  ```
  $paymentObject =
  CRM_Core_Payment::
          singleton('test', $paymentProcessor);
  ```

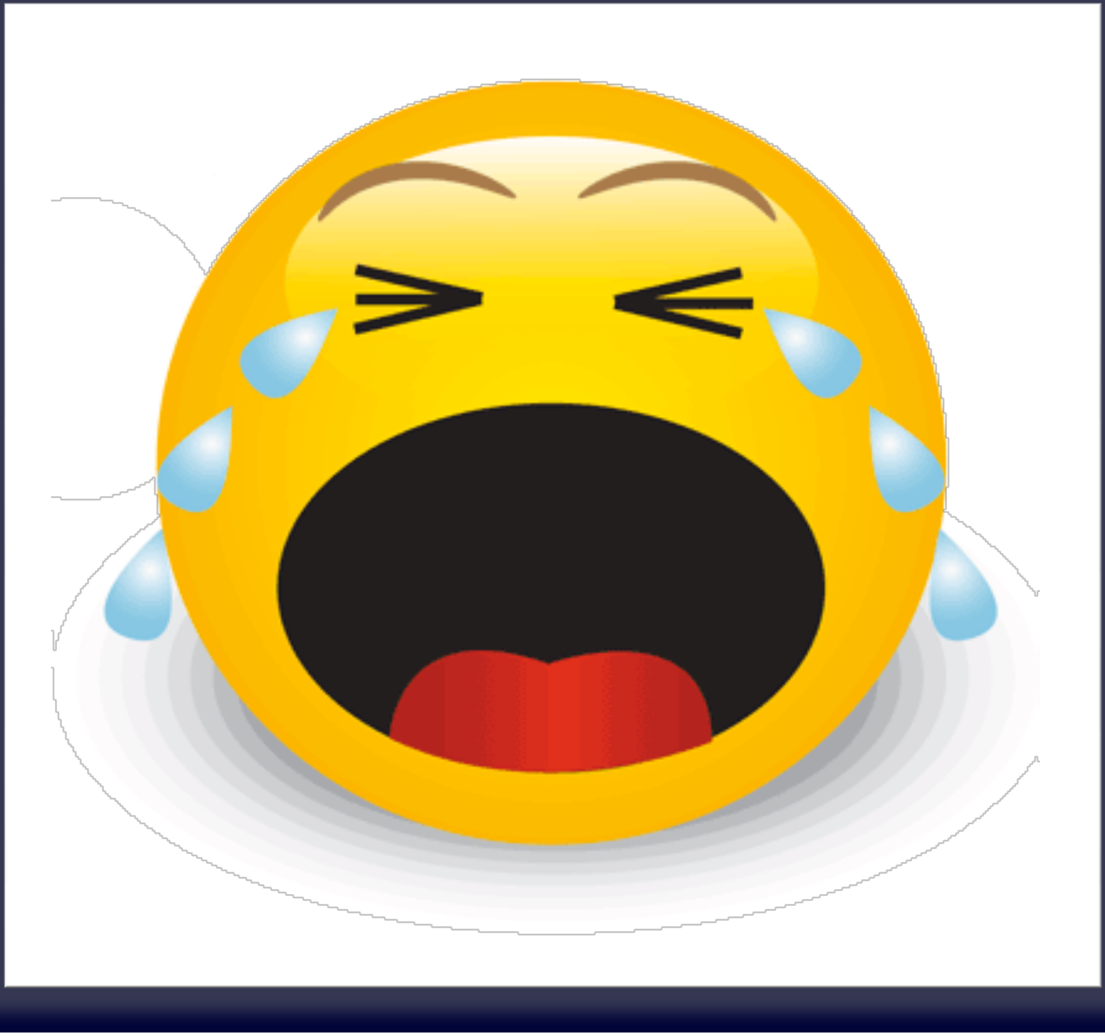
Start Phasing Them Out Now!

New Way

- Civi\Payment\System::singleton()->getById($paymentProcessor)
- Civi\Payment\System::singleton()->getByProcessor($paymentProcessor)

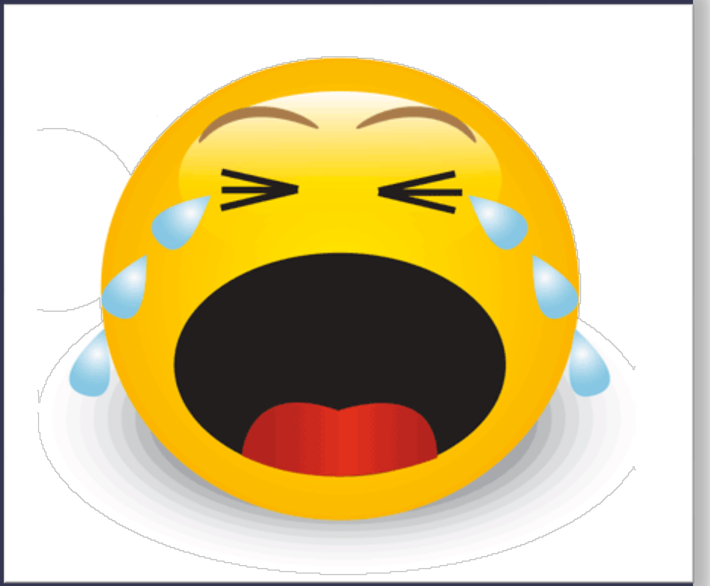
So… surely now we can do off-site transactions via backoffice?

So… surely now we can do off-site transactions via backoffice?



So… surely now we can do off-site transactions via backoffice?

Uh… no - Because the Paypal assumption was so heavily written into the code that backoffice contributions create completed contributions or they get rolled back. In front-end forms the pathways for onsite & offsite transactions after form submission are heavily forked & hard to read...



A Better Approach (IEHO)

- A strong tested API function to create pending transactions for ANY transaction
- When the outcome is known (by whatever means) a function to update this status to completed or failed
- The flow would be the same for on-site AND off-site

A Better Approach (IEHO)

- Could it be the same for pay-later? (one to ponder) **Pay Later** has extra complications – but are they really also relevant to incomplete?
- Could Pay Later be implemented as a processor?

Recurring Transactions & Tokens

Need to STOP writing these per processor!

- contribution.repeattransaction (exists) to create a new transaction copying soft_credits, line_items etc (CRM-15996)
- paymenttoken extension has started to create common storage, api & views access for tokens – but core would make more sense

Omnipay = Open Source Framework for Payment Processors

- allows Omnipay Gateways to be plugged in
- provides mechanism for TransparentRedirect and offsite POST
- implements IPNs provided there is a transaction identifier (current problem with Mollie)

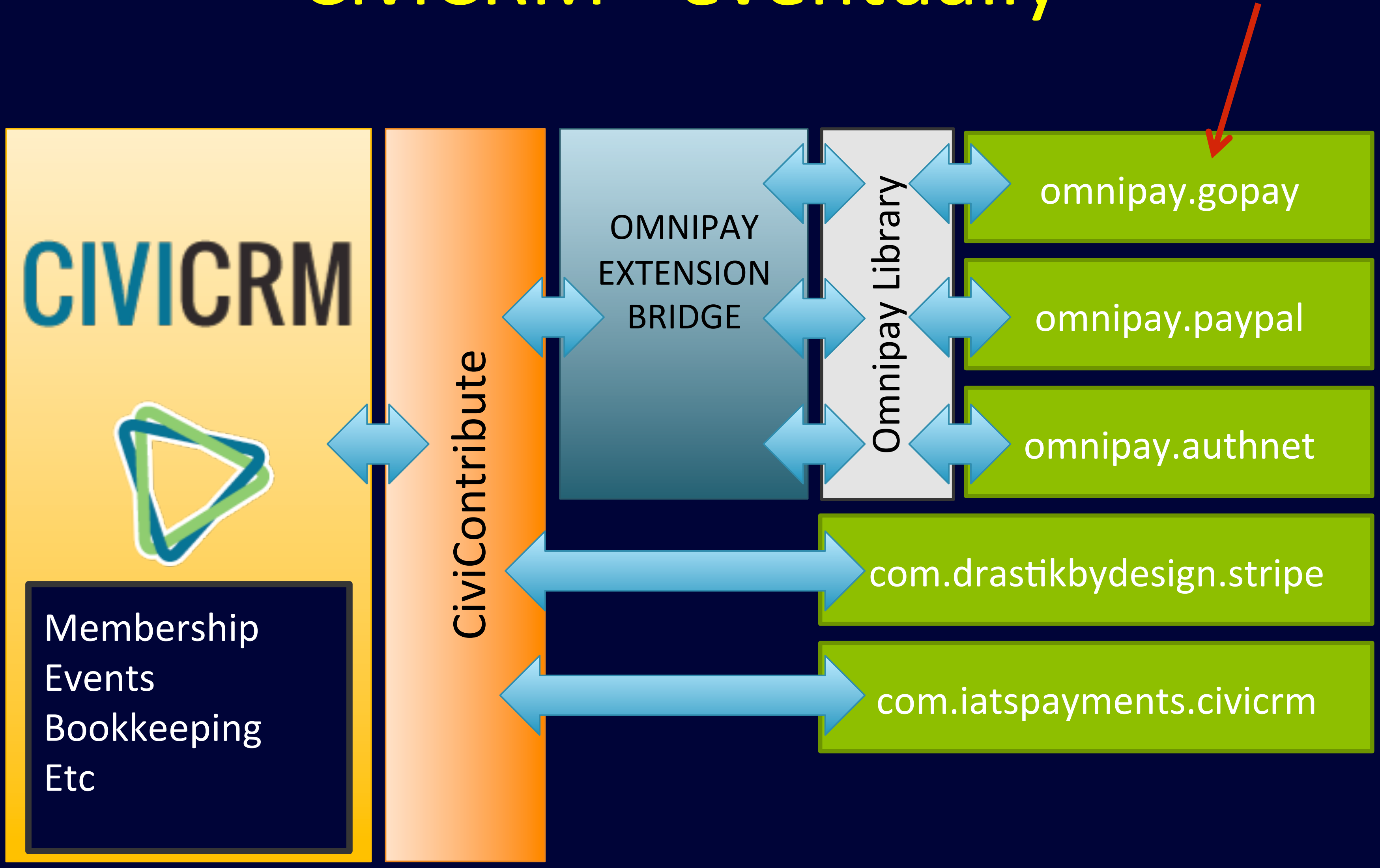
https://github.com/eileenmcnaughton/

nz.co.fuzion.omnipaymultiprocessor.git

Convince me - why not Omnipay?

- of the 30 Payment Processors out there: 3 currently integrated into CiviCRM via OmniPay
- Omnipay is a good model for fairly simple processors - about 80% of those in the wild could be in the Omnipay stable
- but complex ones like Stripe or iATS Payments may not fit - but that’s ok - they can remain Extensions
- not sure about PayPal Express?

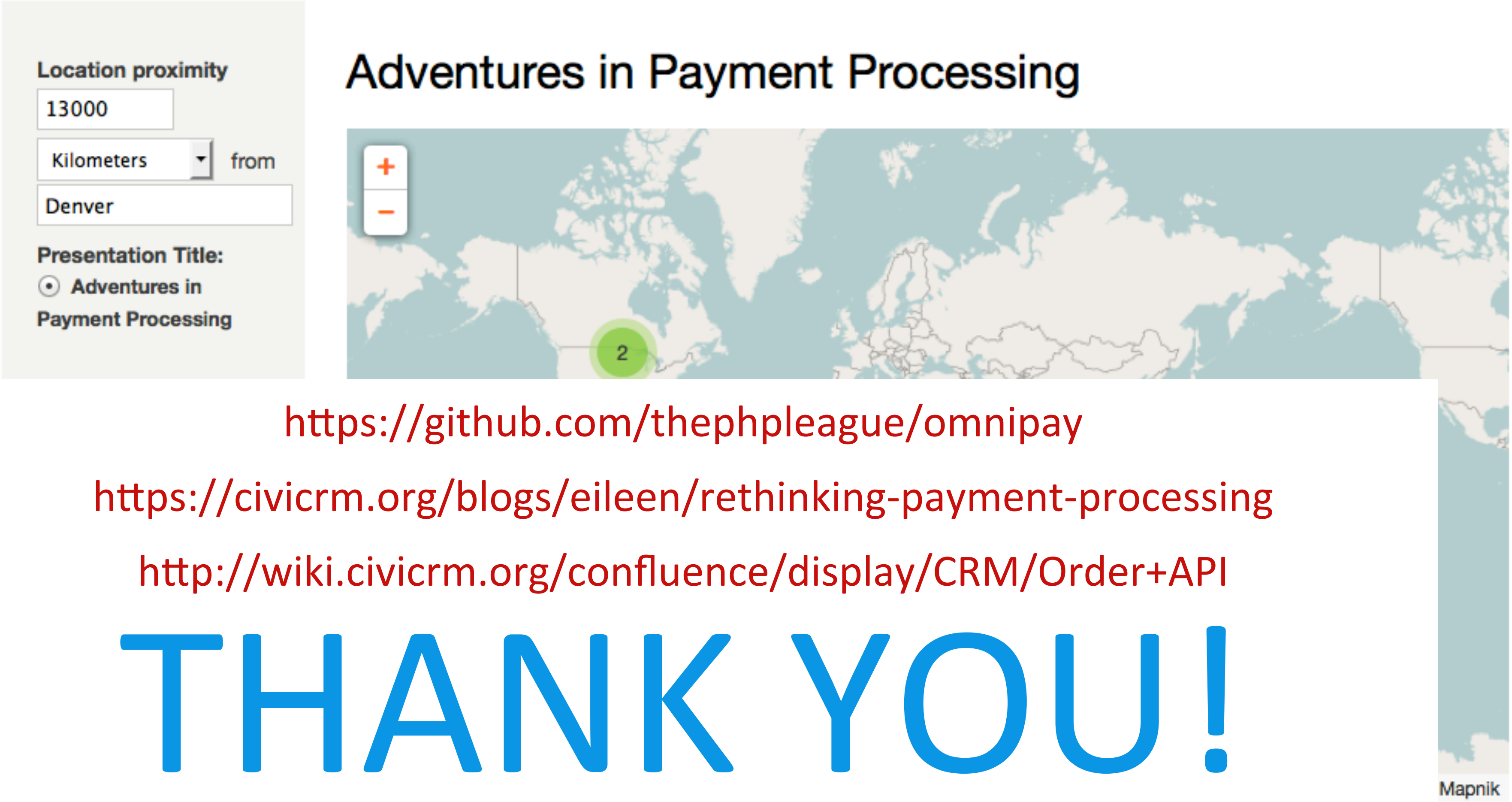
CiviCRM - eventually



Park Canada: Fire plays many important ecological roles in ecosystems - it is beneficial and necessary to maintain health and diversity!

Your CiviCRM needs you!

- Can CiviCRM afford NOT to provide a better framework that would support new and advanced Payment Processor functionality?
- Is there anything more important to so many organizations than the ability to process a variety of live payments across all pathways?



Location proximity

13000

Kilometers from

Denver

Presentation Title:

● Adventures in Payment Processing

Adventures in Payment Processing

https://github.com/thephpleague/omnipay

https://civicrm.org/blogs/eileen/rethinking-payment-processing

http://wiki.civicrm.org/confluence/display/CRM/Order+API

THANK YOU!

Name	Country	Location	Proximity	Group Name	cid	nid
Eileen	New Zealand	POINT (174.7636635 -41.265355)	12,108	Adventures in Payment Processing	57	201
adixon	Canada	POINT (-79.4306098 43.6613773)	2,157	Adventures in Payment Processing	56	200
KarinG	Canada	POINT (-114.2162111 51.0141351)	1,443	Adventures in Payment Processing	2	198