Payment Processing with

ActiveMerchant

Cody Fauser

Payment Processing with

ActiveMerchant

Cody Fauser

Background

shopify

The perfect online store

hosted e-commerce. mere minutes to set up

Snowdevil – Your Shopping Cart

http://www.snowdevil.ca/cart

This is a **Shopify** demo store. We do not have the products advertised here and we don't really charge money upon order completion.

Search   About Us   Shopping Cart **(1 item)** 🛒

Snow Devil

- **Home**
- **Snowboards**
- **Bindings**
- **Goggles**
- **All**
- **Blog**

Search...

Blowout Deals
Arbor: A-Frame 250$
Arbor: Legacy 199$
Carbon 900 bindings 180$

Shopping Cart

Review the items in your cart, make any changes, then proceed to the checkout, or continue shopping for more products.

Check out with PayPal
The safer, easier way to pay

Proceed to Checkout

Product	Price	Quantity	Extension	
⊘ Never Summer: Premier	$399.00	1	$399.00	remove
		Subtotal:	**$399.00** USD	

Update Cart

You may also like...

This store is only for demonstration purposes. No items will be shipped

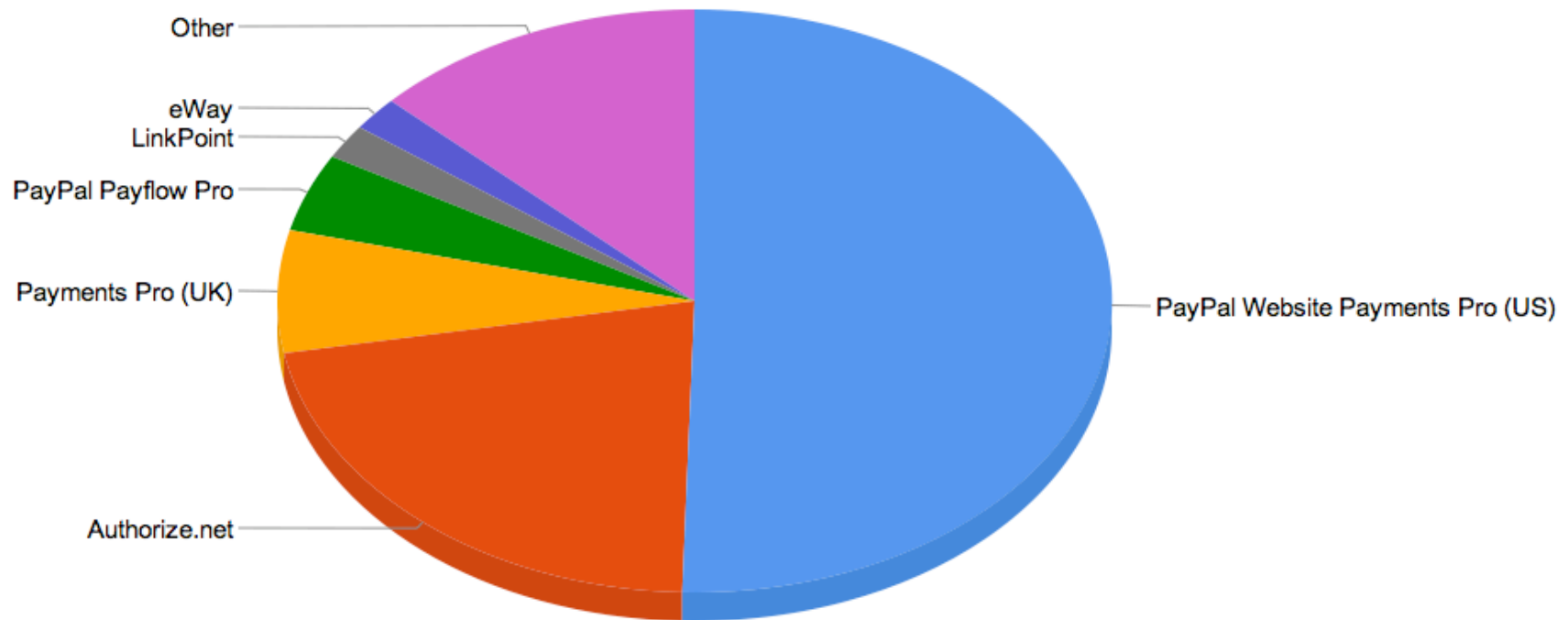All rights reserved.

shopify

The perfect online store

hosted e-commerce. mere minutes to set up

tesla motors

Shopify Gateway Use

Authorizations per Month
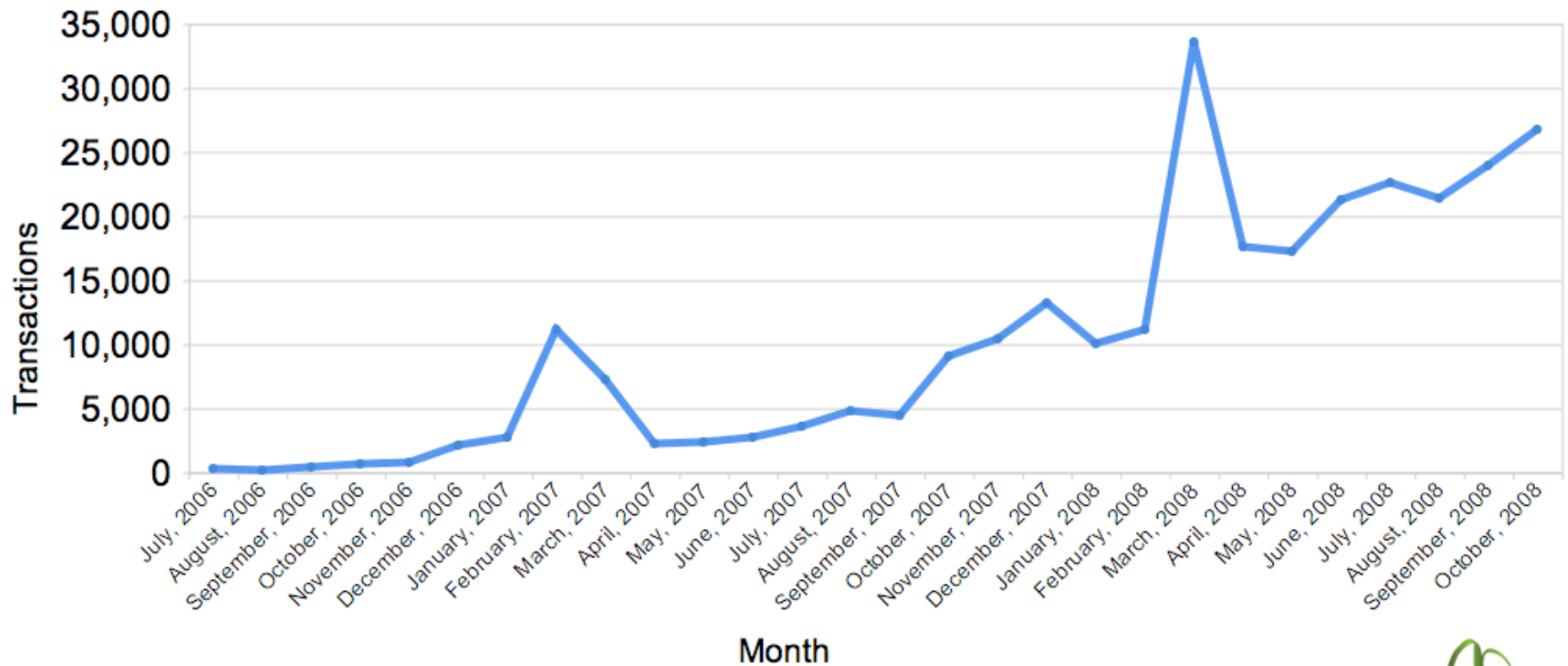
Goal

Simple uniform API

Shopify

↓
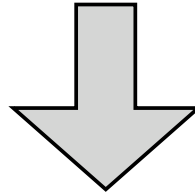
ActiveMerchant

Gateway

Payflow — **PayPal** — **Braintree**

↓
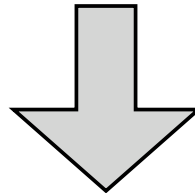
Payment Gateways

The universal translator (also referred to as a "U.T." or translator circuit) is a device used to decipher and interpret alien languages into the native language of the user

- http://memory-alpha.org/en/wiki/Universal_translator

Terminology

Cardholder

Person you want to get money from

Authorization

Verifies and reserves funds on card

Capture

Marks authorization for settlement

Settlement

Transfers authorized amount to your bank

Sale or Purchase

Authorization and capture combined

Address Verification Service (AVS)

Matches street number and zip code to info on file at issuing bank

Card Security Code (CID, CVC, CVV2)

3 or 4 digit security code printed on the back of the card

How?

Authorization

Request

Your app

Response

Processing
Network

Payment
Gateway

Capture

Your App

Request →

← **Response**

Payment
Gateway

Processing
Network

Settlement
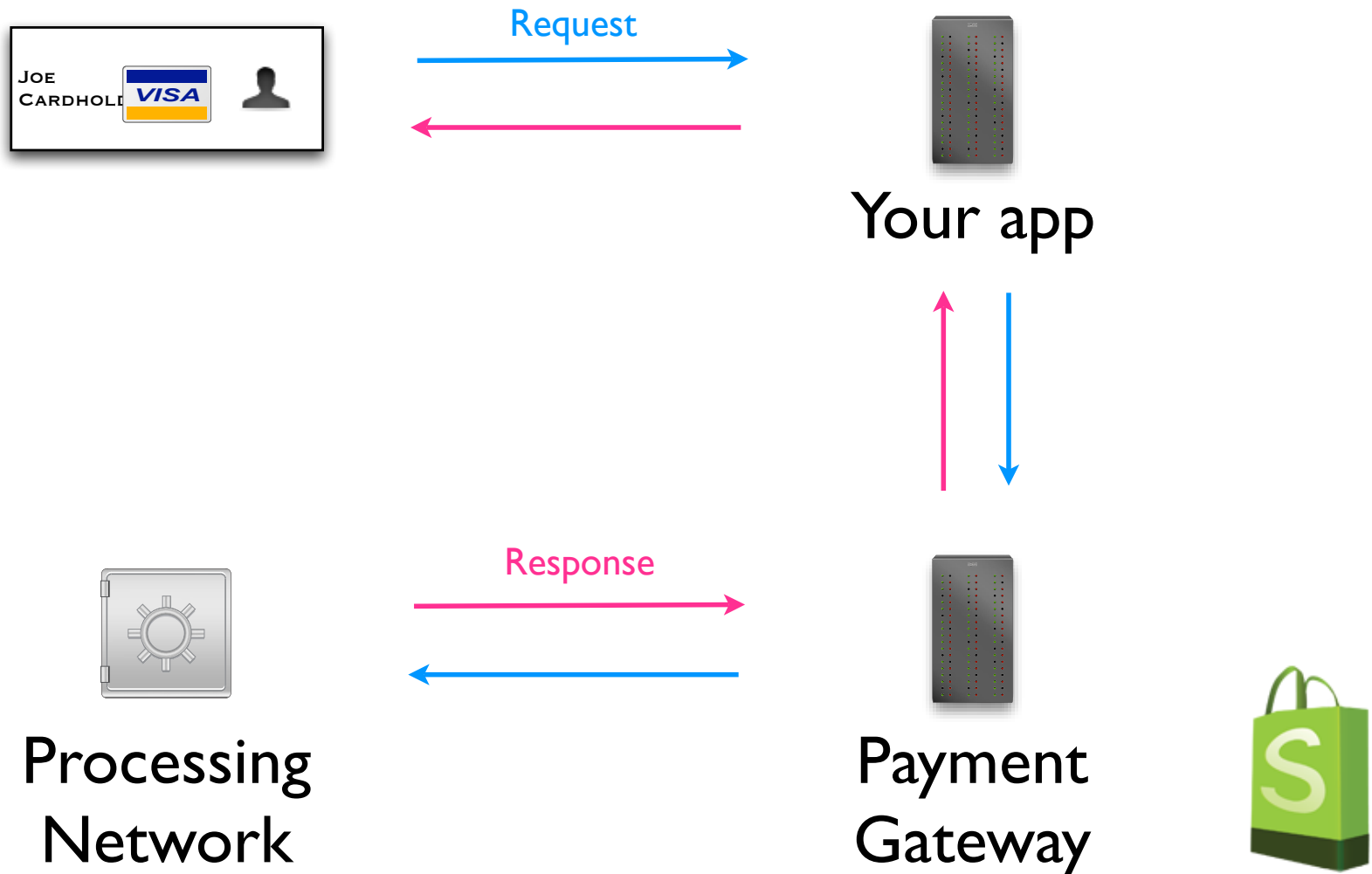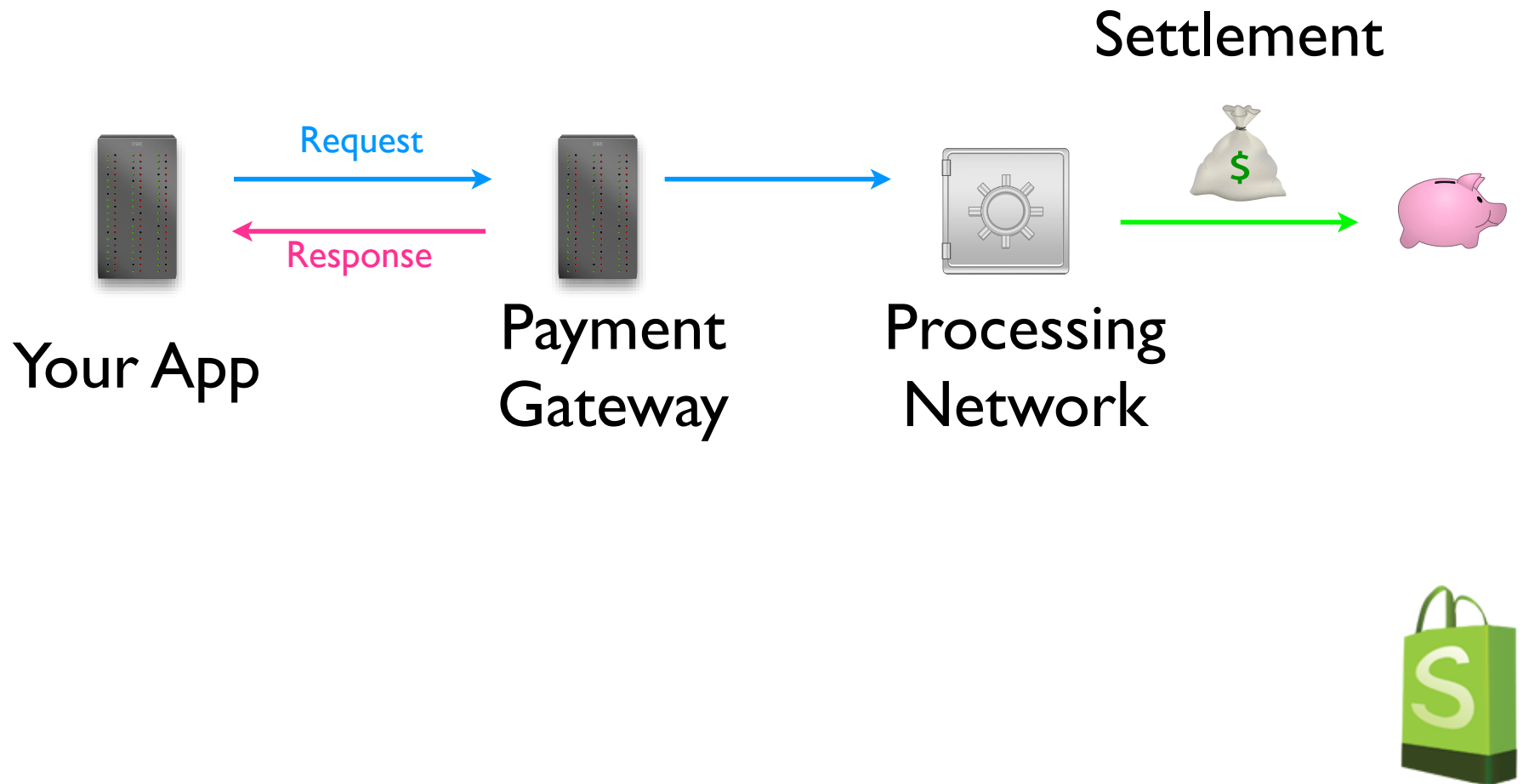
```ruby
card = CreditCard.new(
    :first_name        => 'Tobias',
    :last_name         => 'Luetke',
    :number            => '5473000000000007',
    :month             => '11',
    :year              => '2009',
    :verification_value => '234'
)
```

```ruby
gateway = MonerisGateway.new(
        :login    => 'store1',
        :password => 'yesguy'
      )

response = gateway.purchase(1000, card, :order_id => 'voices-1000')

response.success?      # => true
response.message       # => "Approved"
response.authorization # => "451477-0_7;voices-1000"
```

```ruby
gateway = BraintreeGateway.new(
        :login    => 'demo',
        :password => 'password'
      )

response = gateway.purchase(1000, card, :order_id => 'voices-1000')

response.success?        # => true
response.message         # => "This transaction has been approved"
response.authorization   # => "854203517"
```

```ruby
gateway = TrustCommerceGateway.new(
        :login    => 'TestMerchant',
        :password => 'password'
      )

response = gateway.purchase(1000, card, :order_id => 'voices-1000')

response.success?      # => true
response.message       # => "The transaction was successful"
response.authorization # => "025-0013788895"
```

```ruby
gateway = VerifiGateway.new(
        :login    => 'demo',
        :password => 'password'
      )

response = gateway.purchase(1000, card, :order_id => 'voices-1000')

response.success?       # => true
response.message        # => "Transaction was Approved"
response.authorization  # => "854207614"
```

```ruby
auth = gateway.authorize(1000, card, :order_id => 'voices-1001')

if auth.success?
  gateway.capture(1000, auth.authorization)
else
  raise "hell"
end
```

Complete my purchase

All transactions are secure and encrypted, and we never store your credit card information. To learn more, please view our privacy policy.

First Name

Cody

Last Name

Fauser

Credit Card Number ✓

5473000000000007

Expiration Date

1 – January   2013

Card Security Code 213   what is this?

```ruby
def pay
  @credit_card = CreditCard.new(params[:credit_card])

  if @credit_card.valid?
    @purchase = gateway.purchase(@amount, @credit_card)

    if @purchase.success?
      redirect_to :action => "confirmation"
    else
      render :action => "pay"
    end
  else
    render :action => "pay"
  end
end
```

Benefits

Rich credit card object

```ruby
card = CreditCard.new(
        :first_name       => 'Tobias',
        :last_name        => 'Luetke',
        :number           => '4242424242424242',
        :month            => '11',
        :year             => '2006',
        :verification_value => '234'
      )

card.name                    # => "Tobias Luetke"
card.display_number          # => "XXXX-XXXX-XXXX-4242"
card.valid?                  # => false
card.type                    # => "visa"
card.errors.on(:year)        # => "expired"
card.errors.full_messages    # => ["Year expired"]
card.expired?                # => true
```

Uniform AVS & CVV responses

Risk Assessment

Credit card ✓

Street address ✓

Zip code ✓

IP Address ✓

Payment Method

XXXX–XXXX–XXXX–6062

Processed by Authorize.net

```ruby
response = gateway.purchase(1000, card)

response.avs_result["message"]
# => "Street address and 9-digit postal code match."
response.avs_result["code"]         # => "X"
response.avs_result["street_match"] # => "Y"
response.avs_result["postal_match"] # => "Y"

response.cvv_result["message"]      # => "Match"
response.cvv_result["code"]         # => "M"
```

Community

Extensible

Production Level Code

Thank You!

Questions?

Resources

http://activemerchant.org