



Developer Workshop

Stripe Billing

October 23, 2019



Agenda

1. Quick intros
2. About the Stripe platform
3. Why Stripe Billing
4. Demo
5. Hands-on exercises
6. Q&A



Tara Seshan

Product Manager,
Stripe Billing

@tarstarr



Chris Trudeau

Developer Relations,
Stripe Billing

@trudeaucj

Quick Intros

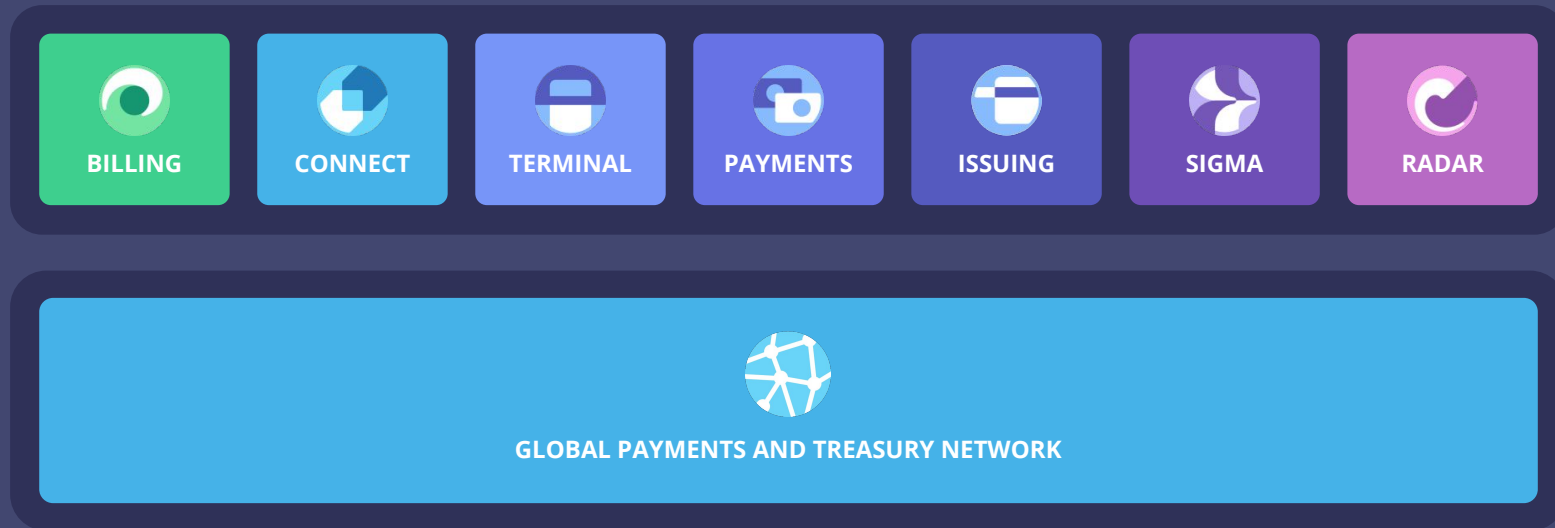
*

Your name

+

What do you want to get out of today

Integrated and unified technology stack for global money movement





Introduction

A developer-friendly platform for
managing subscriptions and invoices.

Why Stripe Billing



Move faster



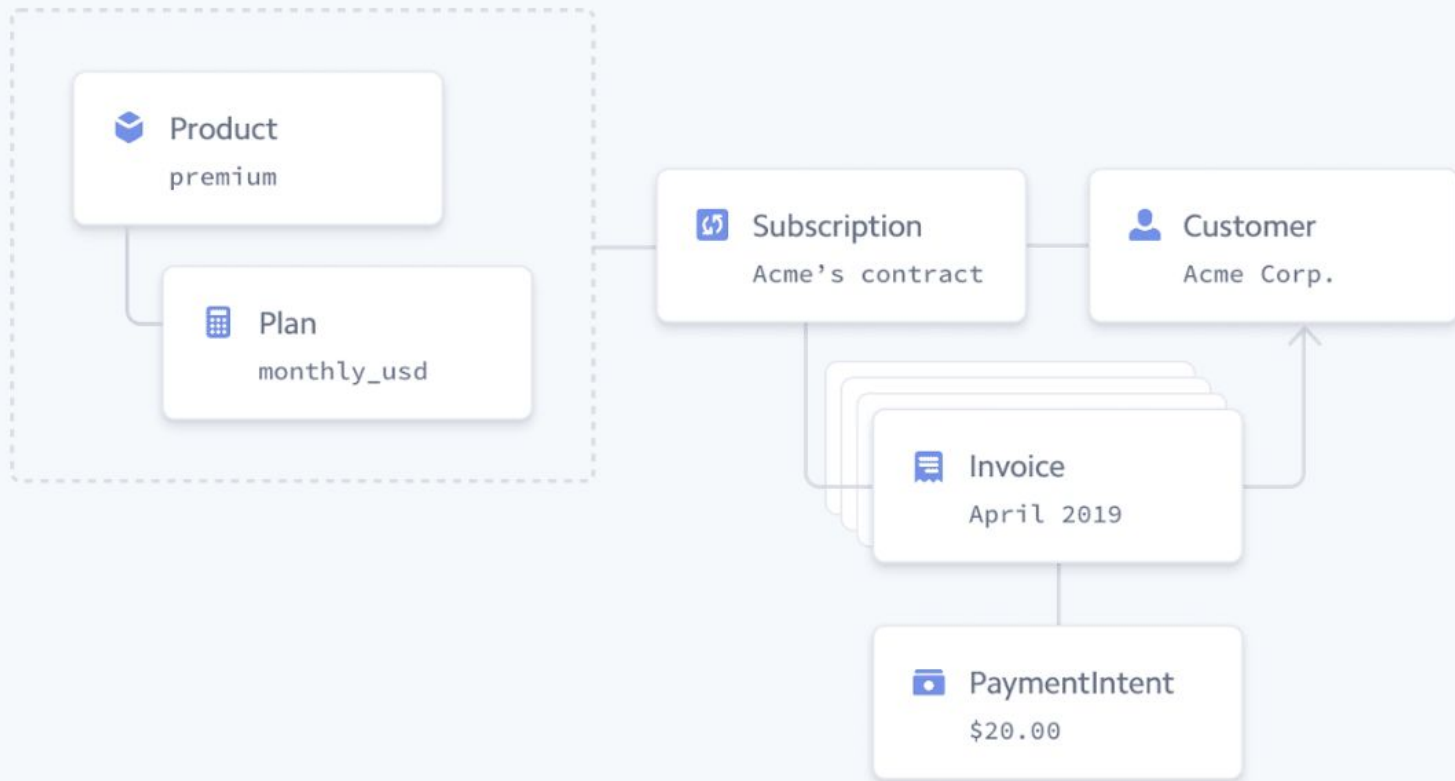
Grow revenue



Scale globally

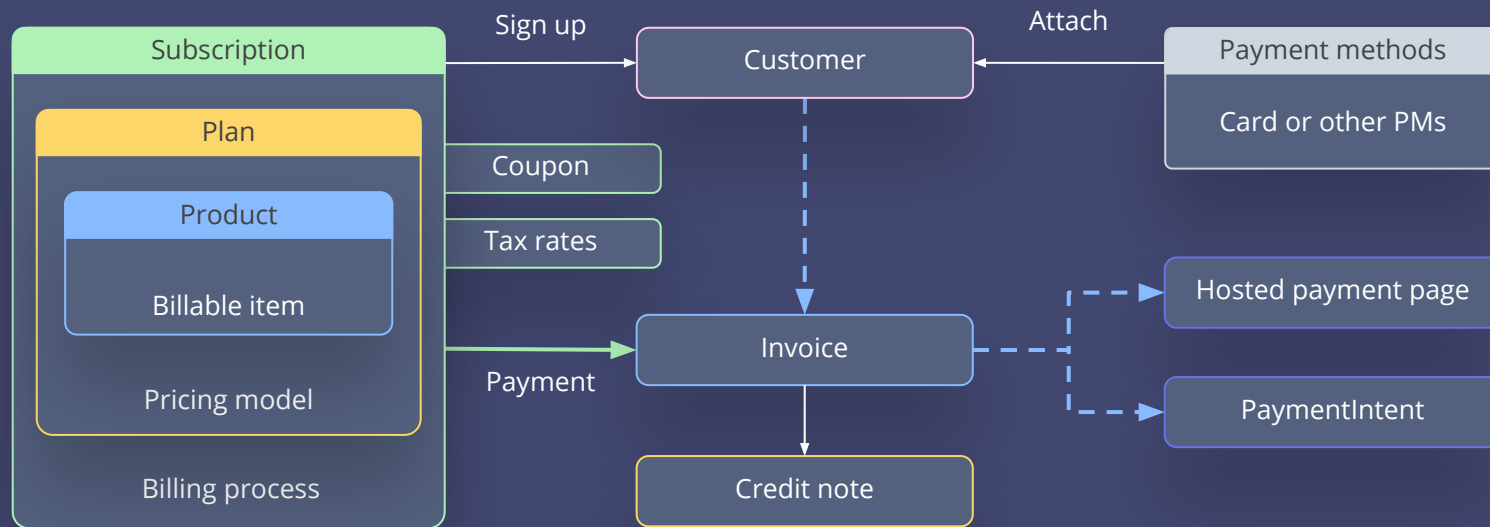


Building blocks





Under the hood



—→ Automatic charge on subscription

- - -→ One-off invoice

Features



ACCEPT ORDER

- From Dashboard
- Stripe.js and Elements
- Stripe Checkout
- Billing API
- Mobile SDK
- SFDC integration

BILLING & INVOICING

- One-off billing
- Recurring billing
- Flat rate billing
- Per seat billing
- Metered billing
- Tiered pricing
- Invoice delivery
- Hosted invoice pages
- Tax rate modelling & integrations

COLLECT PAYMENT

- Credit cards
- ACH and Wire
- ACH auto-reconciliation
- iDEAL
- Bancontact
- Giropay
- SOFORT
- Apple Pay, Google Pay
- Card updater
- Smart retries
- Dunning emails

ACCOUNT & REPORT

- Analytics dashboard
- Custom reporting
- Revenue Recognition (beta)
- NetSuite integration
- Salesforce integration
- Xero integration
- Third-party integrations



Latest features

SMART RETRIES

Automated recovery features powered by Machine Learning.

FLEXIBLE INVOICE NUMBERING

You can choose to number invoices. You can also override sequencing and reset sequencing.

REVENUE RECOGNITION

Revenue recognition tooling that allows you to move from cash accounting to accrual accounting

STRONG CUSTOMER AUTHENTICATION

Stripe Billing is SCA compliant for all users and includes configurable SCA Radar rules.



We want your feedback!

Sign up for early access:
billing-betas@stripe.com



@tarstarr



Demo

Code walkthrough



http://bit.ly/billing_repo

What's next?



Pasha

Enter details below

Payment details

Email



Card number

MM / YY

Create customer



Pasha

Subscribe to a plan

Basic

\$5.99

Per user, per month billed monthly

Subscribe to basic

Premium

\$12.90

Per user, per month billed monthly

Subscribe to premium



Resources

- [Billing workshop GitHub repo](#) (incl. this presentation)
- [Billing developer quickstart](#)
- [Webhooks and subscriptions](#)
- [Stripe CLI](#)
- [Testing Stripe Billing](#)
- [Payment Intents APIs](#)
- [Stripe Elements](#)
- [Stripe Samples](#)



Break



Q&A

Workshop Survey

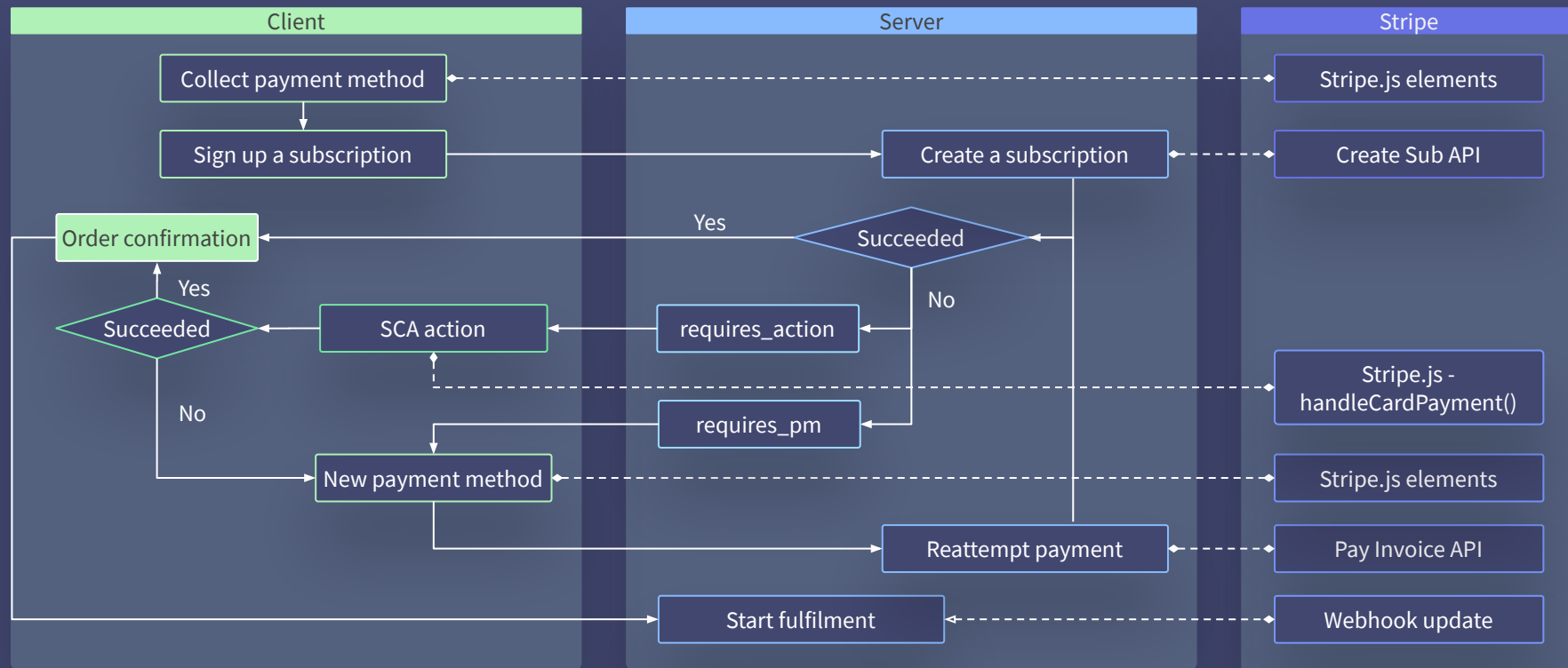


http://bit.ly/billing_survey

Thank you!

APPENDIX

Subscription payment flow





Use case: Apply a discount

A customer is granted a 20% off for the current billing period.

Using Coupon you can easily manage one-off or repeating discounts with either amount or percentage off

```
Stripe::Coupon.create({  
  duration: 'once',  
  percent_off: 20,  
  id: '20OFF',  
})
```

```
Stripe::Subscription.update({  
  customer: '{{CUST_ID}}',  
  items: [{  
    coupon: '20OFF'  
    # Stripe::SubscriptionItem data[0].id  
  }],  
})
```



Use case: Change an existing subscription

A customer initially paid £15.92 per month (1st day of the month) for an individual level 2 dental plan. After a while he/she decided to upgrade to a partner level 4 plan that costs £57.74 per month.

Using the `proration_date` to set when the new plan will start. Stripe will calculate the remaining time/cost on the new plan and the unused time/cost on the old plan

```
Stripe::Plan.create({
  nickname: 'par dental level 4',
  product: '{{PAR_LEVEL_4_PROD_ID}}',
  amount: 5774,
  currency: 'gbp',
  interval: 'month',
  usage_type: 'licensed',
})
```

```
Stripe::Subscription.update({
  items: [{
    plan: '{{PLAN_ID}}'
    id: {{subscription_id}} data[0].id
  }],
  proration_date: 1552262400,
})
```


Use case: Usage-based pricing model

A business wants to pay for all employee's health level 2 plan monthly. Since the number of employees varies month by month, it's difficult to track the monthly payments.

Using Stripe metered billing, you can pass the usage record within the current subscription period, Stripe will calculate the total cost and charge the customer automatically

```
Stripe::Plan.create({  
  nickname: 'biz health level 2',  
  product: '{{BIZ_LEVEL_2_PROD_ID}}',  
  amount: 2000,  
  currency: 'gbp',  
  interval: 'month',  
  usage_type: 'metered',  
})
```

```
Stripe::Subscription.create({  
  customer: '{{CUST_ID}}',  
  items: [{  
    plan: '{{PLAN_ID}}'  
  }],  
})
```

```
Stripe::UsageRecord.create({  
  subscription_item: '{{SI_ID}}',  
  quantity: 120  
  meter_id: 1552744368  
})
```

Use case: Tiered pricing model

A business wants to pay for all employee's health level 2 plan monthly. A varied unit price can be offered based on the number of employees

Number of employees	Price per person
1~50	£20
51~100	£18.5
101~200	£16.5
200+	£15

```
Stripe::Plan.create ({
  nickname: 'biz health level 2 - tiered' ,
  product: '{{BIZ_LEVEL_2_PROD_ID}}' ,
  currency: 'gbp',
  interval: 'month',
  usage_type: 'metered',
  [
    {
      amount: 2000
      up_to: 50
    },
    ...
    {
      amount: 1500
      up_to: 'inf'
    }
  ]
})
```

```
Stripe::Subscription.create ({
  customer: '{{CUST_ID}}' ,
  items: [{
    '{{PLAN_ID}}'
  }],
})
```

```
Stripe::UsageRecord.create ({
  subscription_item: '{{SI_ID}}' ,
  quantity: 120
  meter_id: 1552744368
})
```

stripe