

# 无处不在的新零售

- 用全球化支付迎接新一代用户

- 新零售的3个W
  - What?
  - Why?
  - How? 趋势和预测
- 安卓支付
- 精益求精

# WHAT?

@droidcon | @zhaosiruo | @MobileWorldCongress

所有商业的全面数据化。



ALIBABA'S NEW RETAIL EXPLAINED

# WHY?

@droidcon | @zhaosiruo | @MobileWorldCongress

博采众长。





# HOW?

@droidcon | @zhaosiruo | @MobileWorldCongress





# HOW?

@droidcon | @zhaosiruo | @MobileWorldCongress



# HOW?

@droidcon | @zhaosiruo | @MobileWorldCongress

## 移动化





# HOW?

@droidcon | @zhaosiruo | @MobileWorldCongress

## 自动化



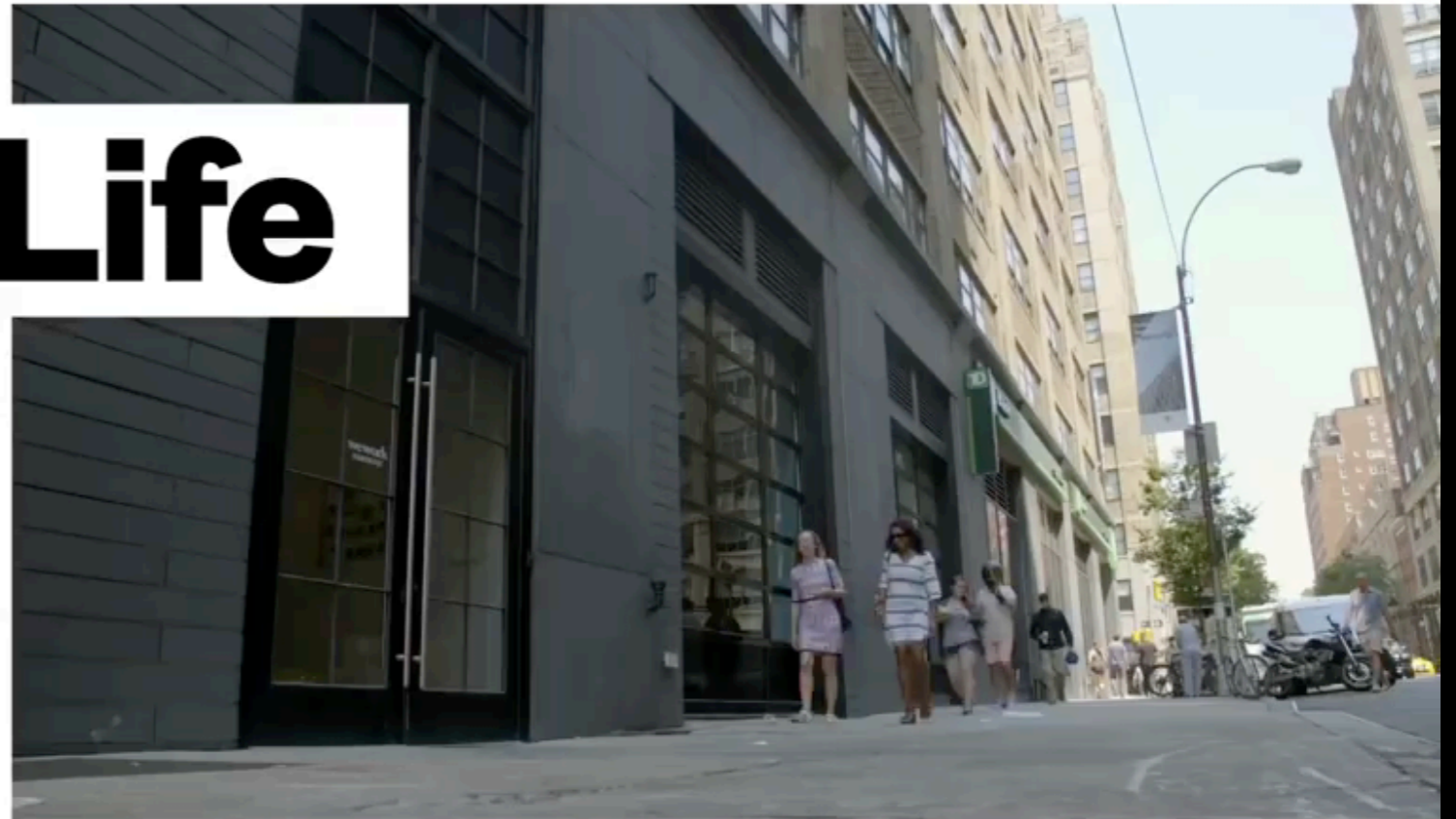
FELLOW ROBOTS@BEVMO

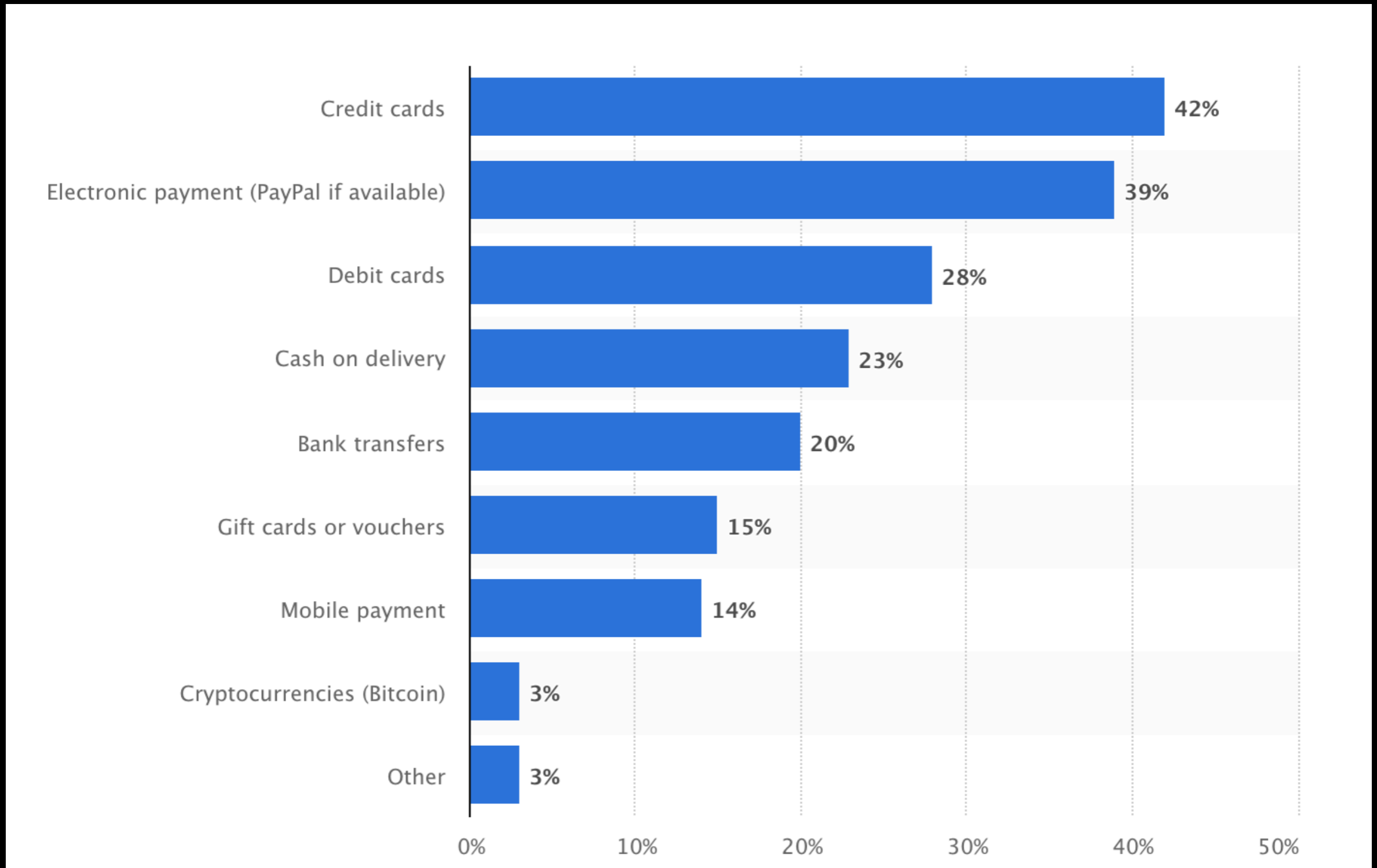


## 广义化

VICE News | HBO

# The WeLife

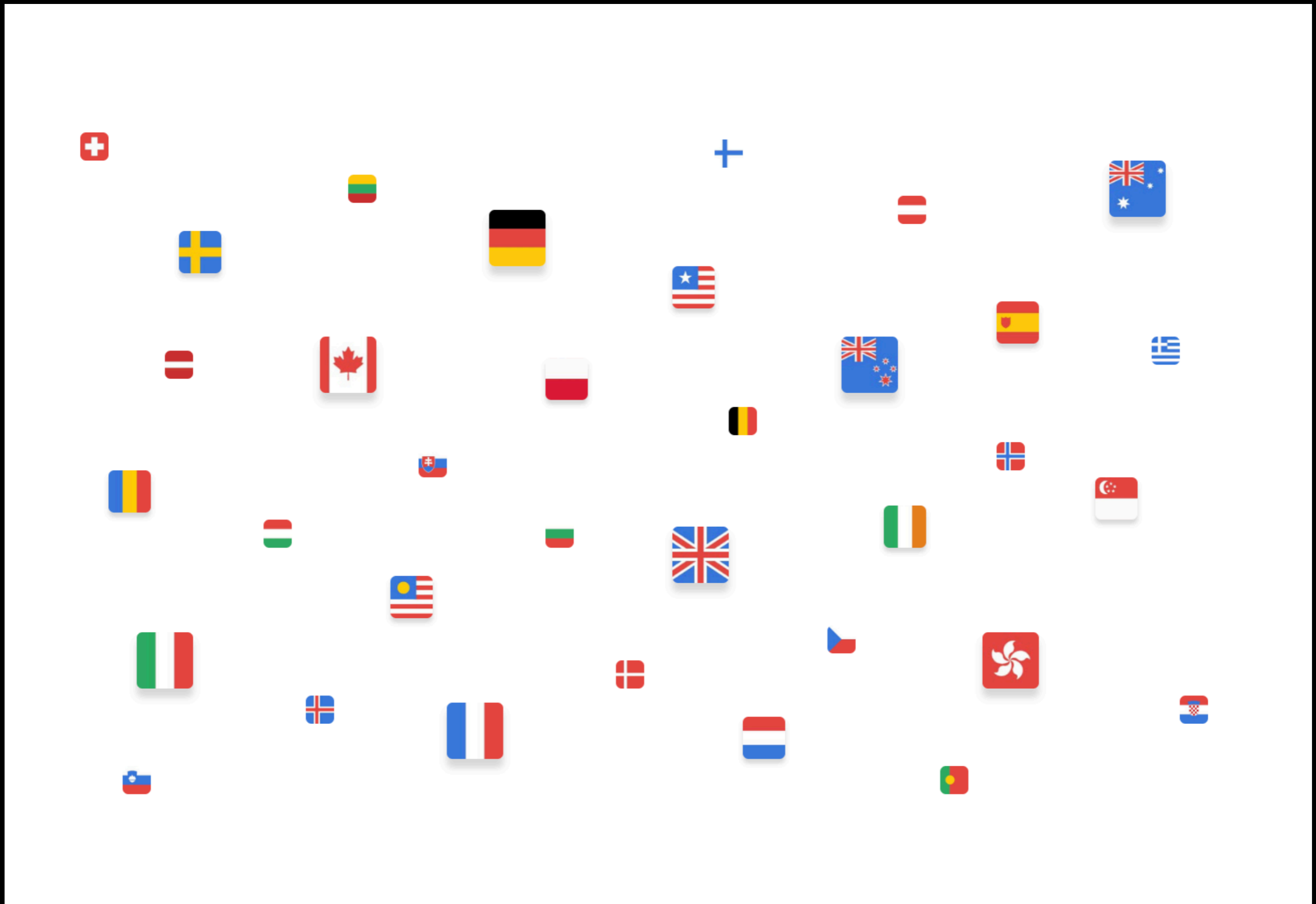




世界范围最受欢迎的线上支付方式 (2017年三月)

# BRAINTREE BY PAYPAL

@droidcon | @zhaosiruo | @MobileWorldCongress





# BRAINTREE BY PAYPAL

@droidcon | @zhaosiruo | @MobileWorldCongress

## CREDIT / DEBIT

### Credit/debit cards

Start with the basics -- accept credit and debit cards of all kinds.



### PayPal

Eliminate friction with checkouts in a click.



### Venmo

Make online purchasing more social.



### Apple Pay

Open shop to an innovative digital wallet.



### Google Pay

More seamless checkouts for users with payment methods saved to their Android device or Google account.



### Visa Checkout

Offer a single sign-on experience for Visa Checkout users.



### Masterpass

Let Masterpass users log in and pay with ease.



### Amex Express Checkout

Help Amex Card Members fly through checkout.

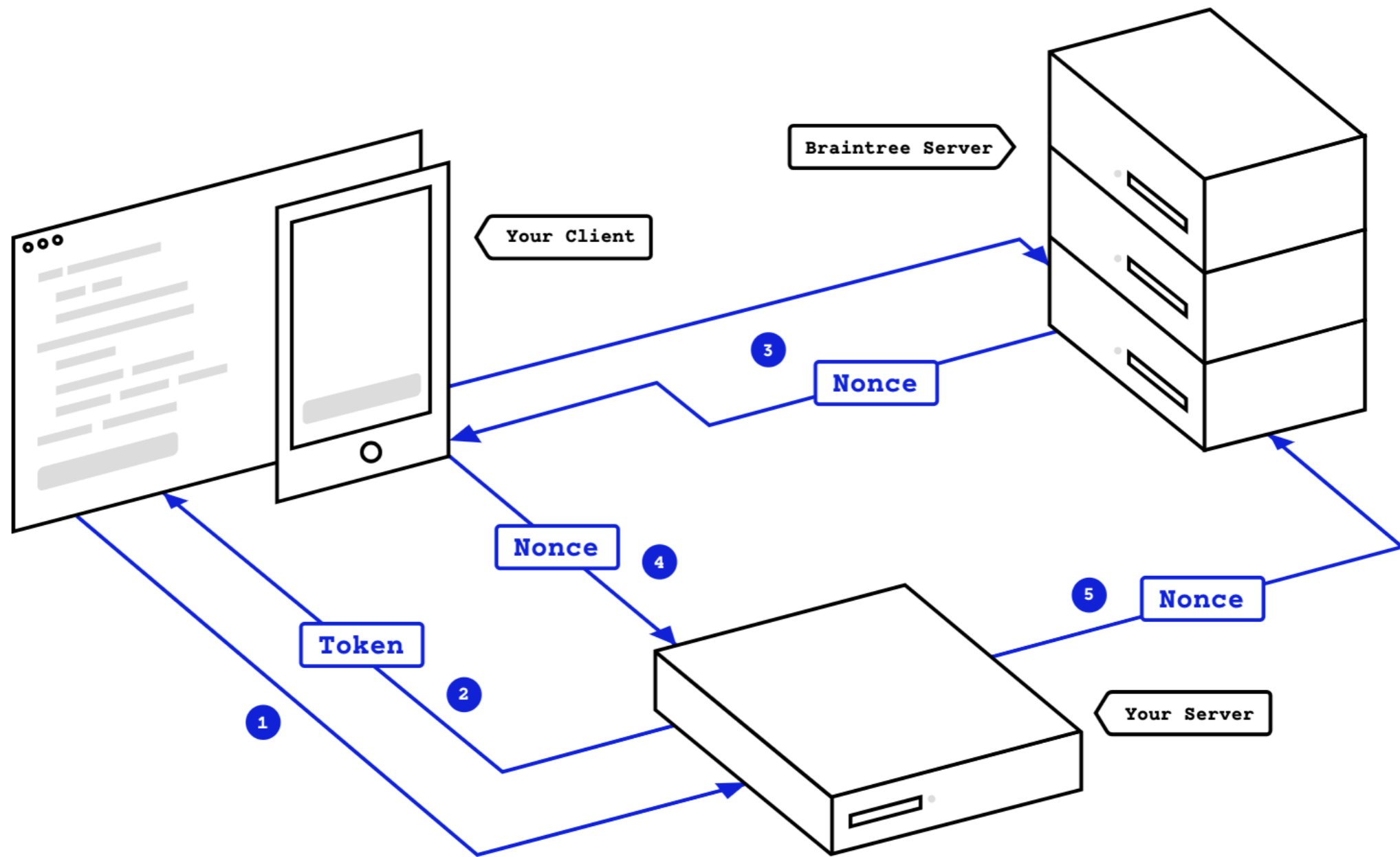


### ACH Direct Debit

Expand your payment options with a cost-effective alternative.

# BRAINTREE BY PAYPAL

@droidcon | @zhaosiruo | @MobileWorldCongress



```
dependencies {  
    ...  
  
    implementation 'com.braintreepayments.api:braintree:2.+'  
    implementation 'com.braintreepayments.api:drop-in:3.+'  
    implementation 'com.loopj.android:android-async-http:1.4.9'  
}
```



# CHECKOUTACTIVITY.JAVA @droidcon | @zhaosiruo | @MobileWorldCongress

---

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

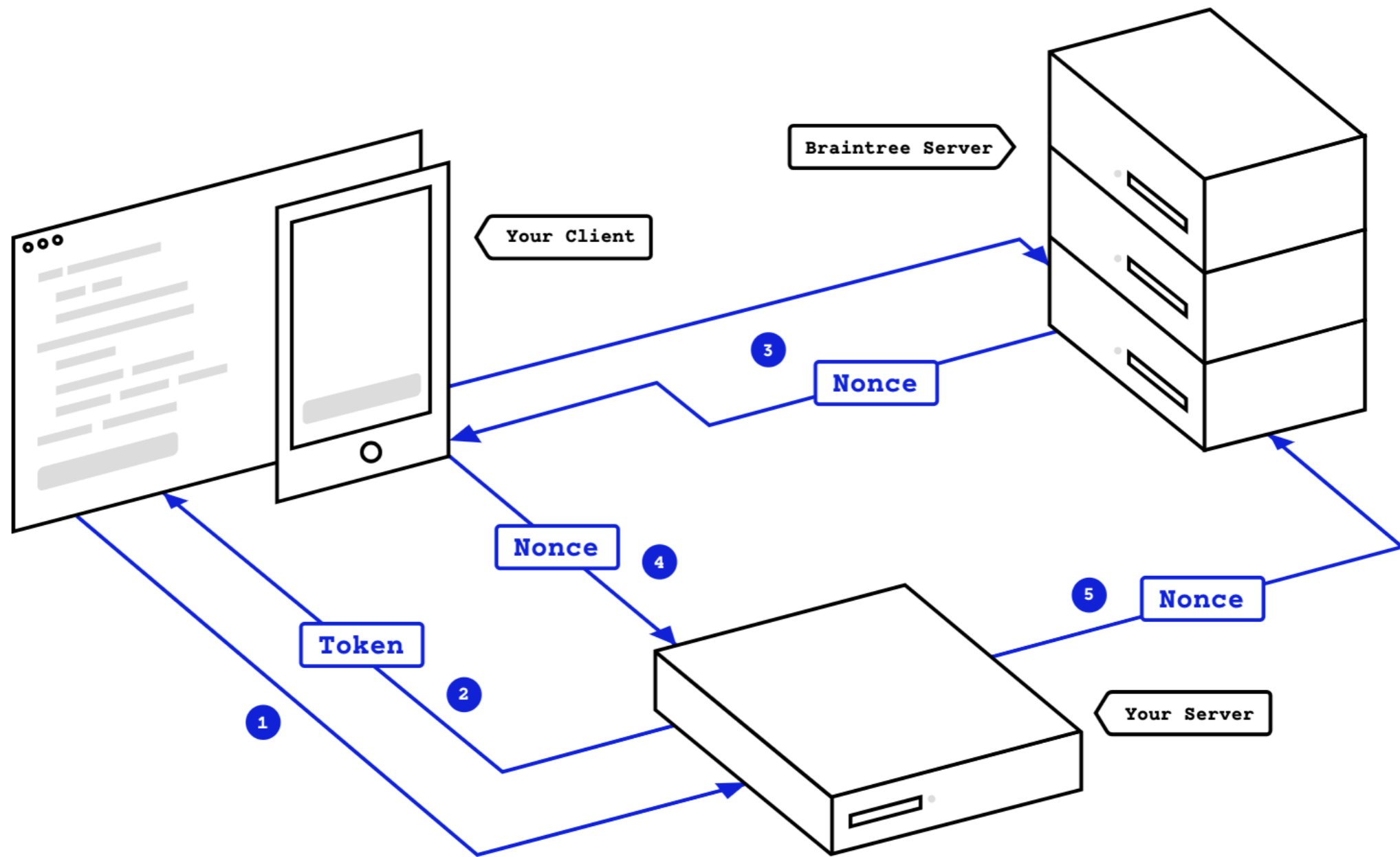
    getClientToken();
    ...
}

private void getClientToken(){
    AsyncHttpClient androidClient = new AsyncHttpClient();
    androidClient.get(PATH_TO_SERVER, new TextHttpResponseHandler() {
        @Override
        public void onFailure(int statusCode, Header[] headers, String
responseString, Throwable throwable) {
            Toast.makeText(getApplicationContext(), responseString,
Toast.LENGTH_LONG).show();
        }

        @Override
        public void onSuccess(int statusCode, Header[] headers, String
responseToken) {
            clientToken = responseToken;
        }
    });
}
```

# BRAINTREE BY PAYPAL

@droidcon | @zhaosiruo | @MobileWorldCongress



```
gem "braintree", "~> 2.89.0"
```



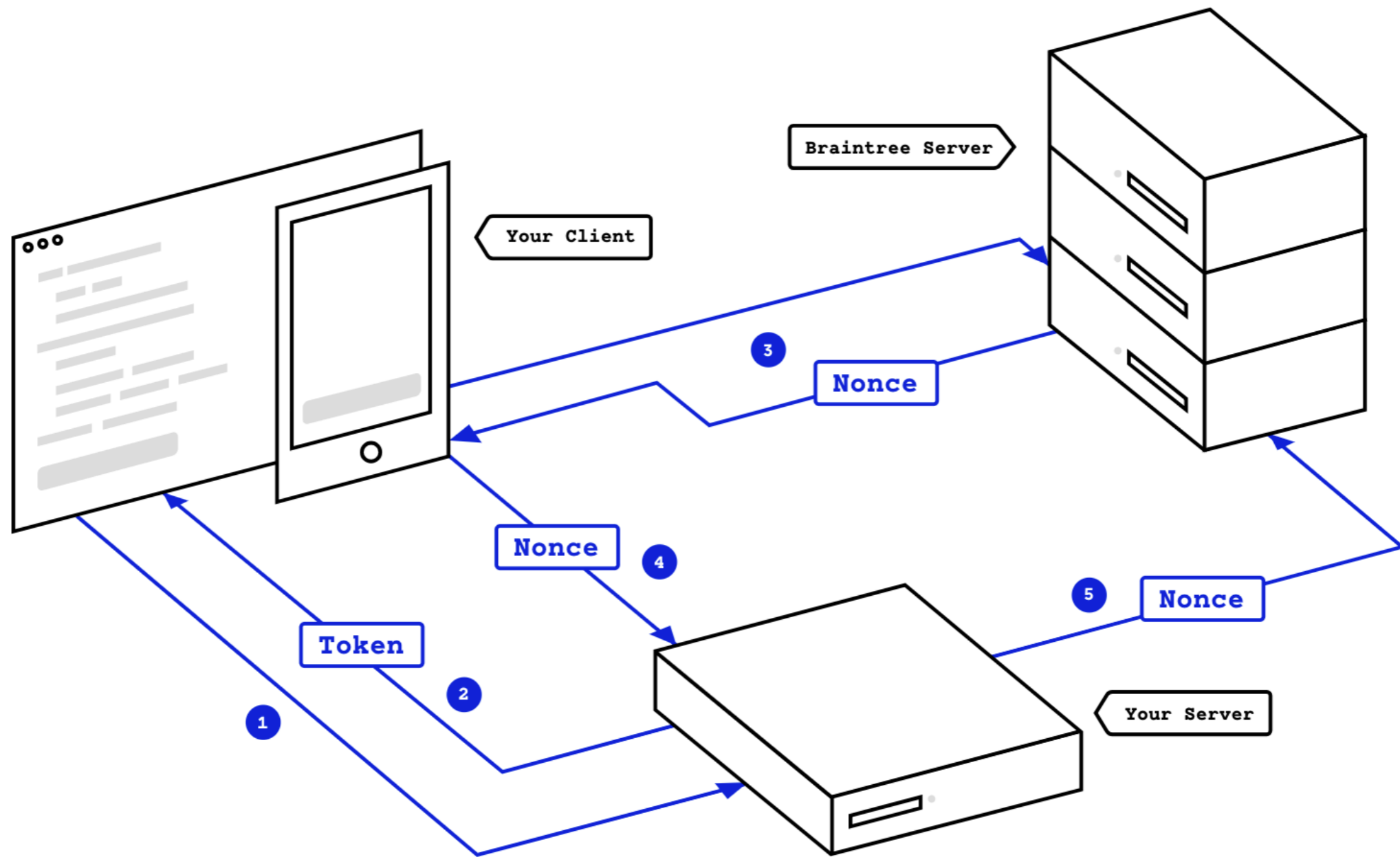
```
gateway = Braintree::Gateway.new(  
  :environment => :sandbox,  
  :merchant_id => "your_merchant_id",  
  :public_key => "your_public_key",  
  :private_key => "your_private_key",  
)
```

```
@client_token = gateway.client_token.generate(  
  :customer_id => a_customer_id  
)
```

```
get "/client_token" do
  gateway.client_token.generate
end
```

# BRAINTREE BY PAYPAL

@droidcon | @zhaosiruo | @MobileWorldCongress



```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    getClientToken();

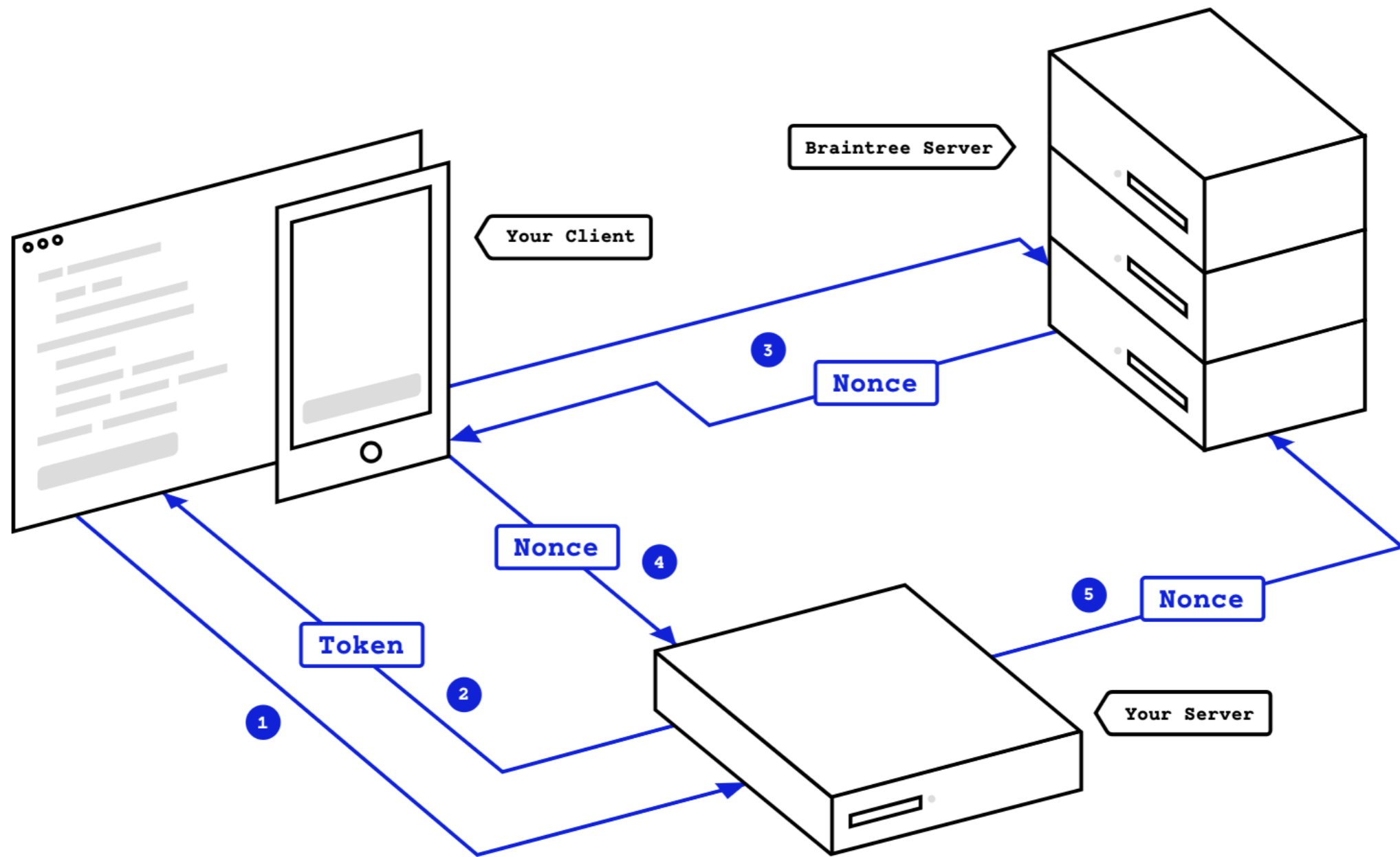
    Button checkoutButton = findViewById(R.id.checkout_button);
    checkoutButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            DropInRequest dropInRequest = new
DropInRequest().clientToken(clientToken);
            startActivityForResult(dropInRequest.getIntent(view.getContext()),
BRAIN_TREE_REQUEST_CODE);
        }
    });
}
```



```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == BRAIN_TREE_REQUEST_CODE) {
        if (resultCode == Activity.RESULT_OK) {
            DropInResult result =
data.getParcelableExtra(DropInResult.EXTRA_DROP_IN_RESULT);
            String paymentNonce = result.getPaymentMethodNonce().getNonce();
            sendPaymentNonceToServer(paymentNonce);
        } else if (resultCode == Activity.RESULT_CANCELED) {
            // the user canceled
        } else {
            // errors
            Exception error = (Exception)
data.getSerializableExtra(DropInActivity.EXTRA_ERROR);
        }
    }
}
```

# BRAINTREE BY PAYPAL

@droidcon | @zhaosiruo | @MobileWorldCongress

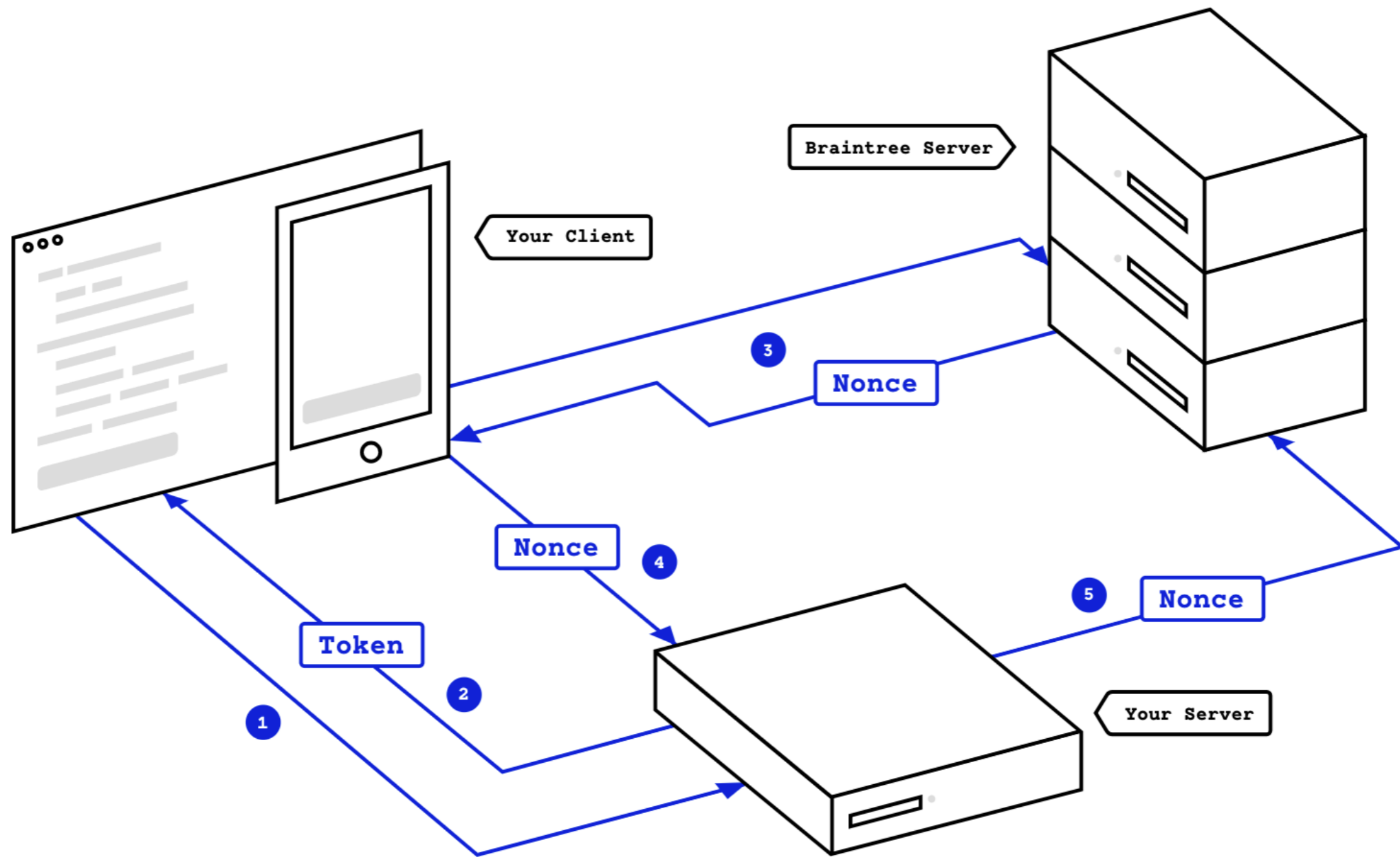


```
private void sendPaymentNonceToServer(String paymentNonce){
    RequestParams params = new RequestParams("NONCE", paymentNonce);
    AsyncHttpClient androidClient = new AsyncHttpClient();
    androidClient.post(PATH_TO_SERVER, params, new TextHttpResponseHandler() {
        @Override
        public void onFailure(int statusCode, Header[] headers, String
responseString, Throwable throwable) {
            Toast.makeText(getApplicationContext(), responseString,
Toast.LENGTH_LONG).show();
        }

        @Override
        public void onSuccess(int statusCode, Header[] headers, String
responseString) {
            Toast.makeText(getApplicationContext(), responseString,
Toast.LENGTH_LONG).show();
        }
    });
}
```

# BRAINTREE BY PAYPAL

@droidcon | @zhaosiruo | @MobileWorldCongress



```
post "/checkout" do
  nonce_from_the_client = params[:payment_method_nonce]
  # Use payment method nonce here...
end
```



```
result = gateway.transaction.sale(  
  :amount => "10.00",  
  :payment_method_nonce => nonce_from_the_client,  
  :options => {  
    :submit_for_settlement => true  
  }  
)
```

安全第一！

顾客至上！

**细节决定成败！**

**谢谢!**