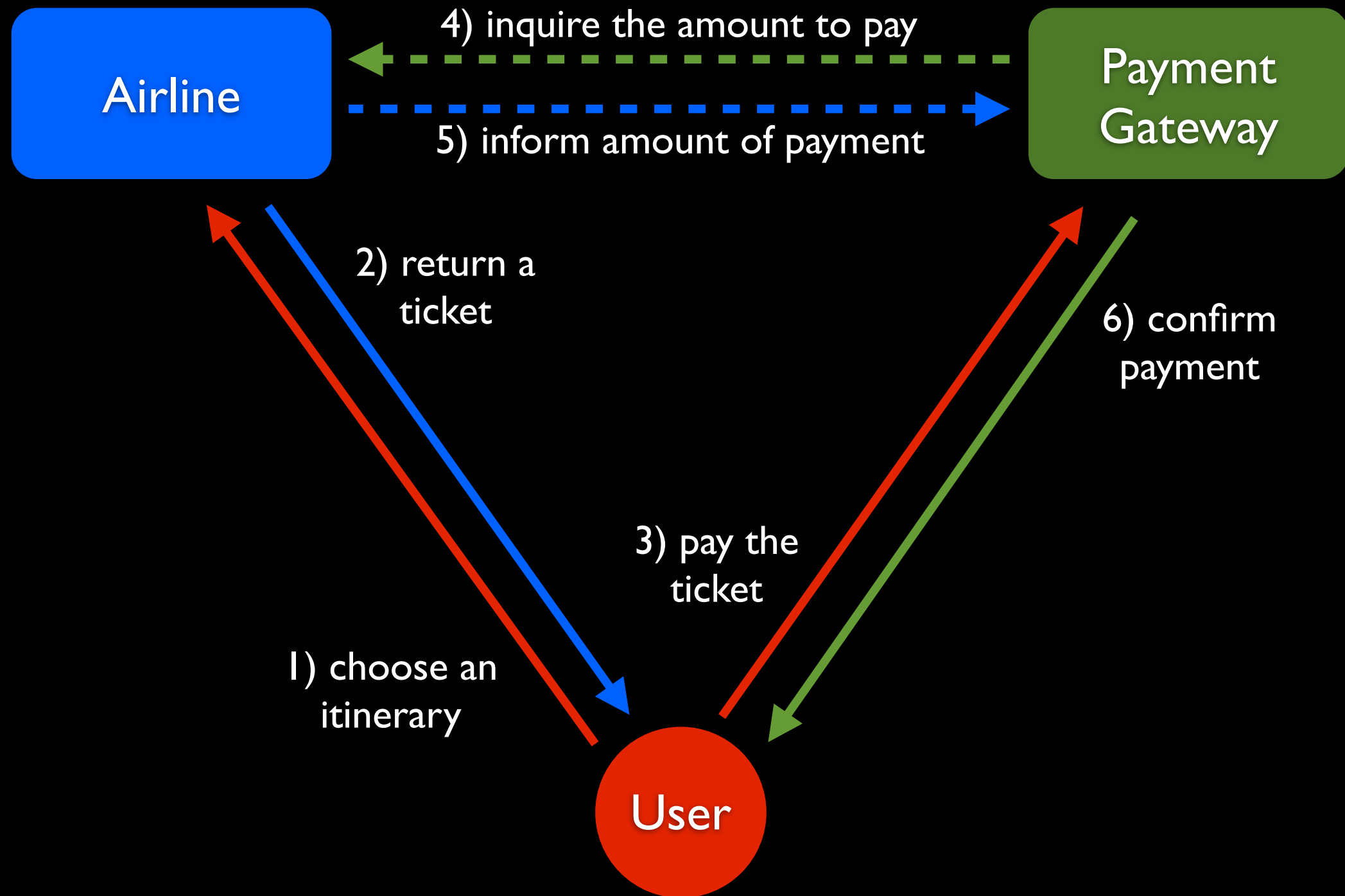


# Booking & Payment

# More Basic Types

- Itinerary (origin, destination, date, etc)
- Traveler (name, gender, passport, etc)
- Ticket (itinerary, travelers, price)

# Booking Procedure



# At Lufthansa

- I visit <http://lufthansa.com> and search and choose an itinerary
- I fill a form specifying how many people are flying this itinerary (e.g. 3 guys)
- Lufthansa takes the itinerary and traveler info, then generates a ticket (a Billable) and presents me a link to EuroPay, e.g.

<http://europay.com/pay?payee=lufthansa&item=1234567890>

- I click the link above and visit EuroPay ...

# At EuroPay

- EuroPay gets the payee and item info from the URL and contacts Lufthansa to get more info about the amount of money I should pay.
- EuroPay accepts Visa, MasterCard, and ElectronicCash.
- I choose to pay with my Visa card and type the card number, expire date, security code, etc.
- EuroPay checks if my Visa card is valid and processes my payment afterwards.
- EuroPay presents me a confirmation message and notifies Lufthansa that the payment is done.
- EuroPay redirects me to Lufthansa for a final confirmation.

# Payment Gateway



```
abstract class Payable
```

```
class CreditCard extends Payable {  
    val cardNumber: String  
    val holderName: String    // as printed on the card  
    val expireDate: Date      // e.g. 2012-12-31  
    val securityCode: String  // usually the 3 digits on the back of the card  
}
```

```
class Visa          extends CreditCard  
class MasterCard    extends CreditCard  
class AmericanExpress extends CreditCard
```

```
class DebitCard extends Payable {  
    val cardNumber: String  
    val holderName: String  
    val expireDate: Date  
    val pin: String  
}
```

```
class ConvenienceCard extends DebitCard  
class ElectronicCash   extends DebitCard  
class UnionPayCard     extends DebitCard
```

```
trait Billable {  
  val id: String    // globally unique identifier  
  val price: Int  
  val payee: String // which company to receive the money?  
}  
// NOTE: a ticket is a Billable!  
  
class PaymentResult {  
  val success: Boolean  
  val message: String    // indicate what's wrong if success == false  
  
  def toXML =  
    <paymentResult>  
      <success>{ success }</success>  
      <message>{ message }</message>  
    </paymentResult>  
}
```

```
abstract class PaymentGateway {
  val url: String // user must visit this url to complete payments

  def pay(item: Billable, visaCard: Visa): PaymentResult = {
    // pre-condition
    assert( item.price > 0 )
    assert( item.payee is valid ) // no money laundry allowed :)
    assert( visaCard is valid ) // check expire date and security code

    // processing

    // post-condition
    assert( visaCard is deducted item.price amount of money )

    new PaymentResult // either success or failure
  }

  def pay(item: Billable, masterCard: MasterCard): PaymentResult = {
    // similar pre-/post-condition for all pay(...) methods here and below
  }
}
```

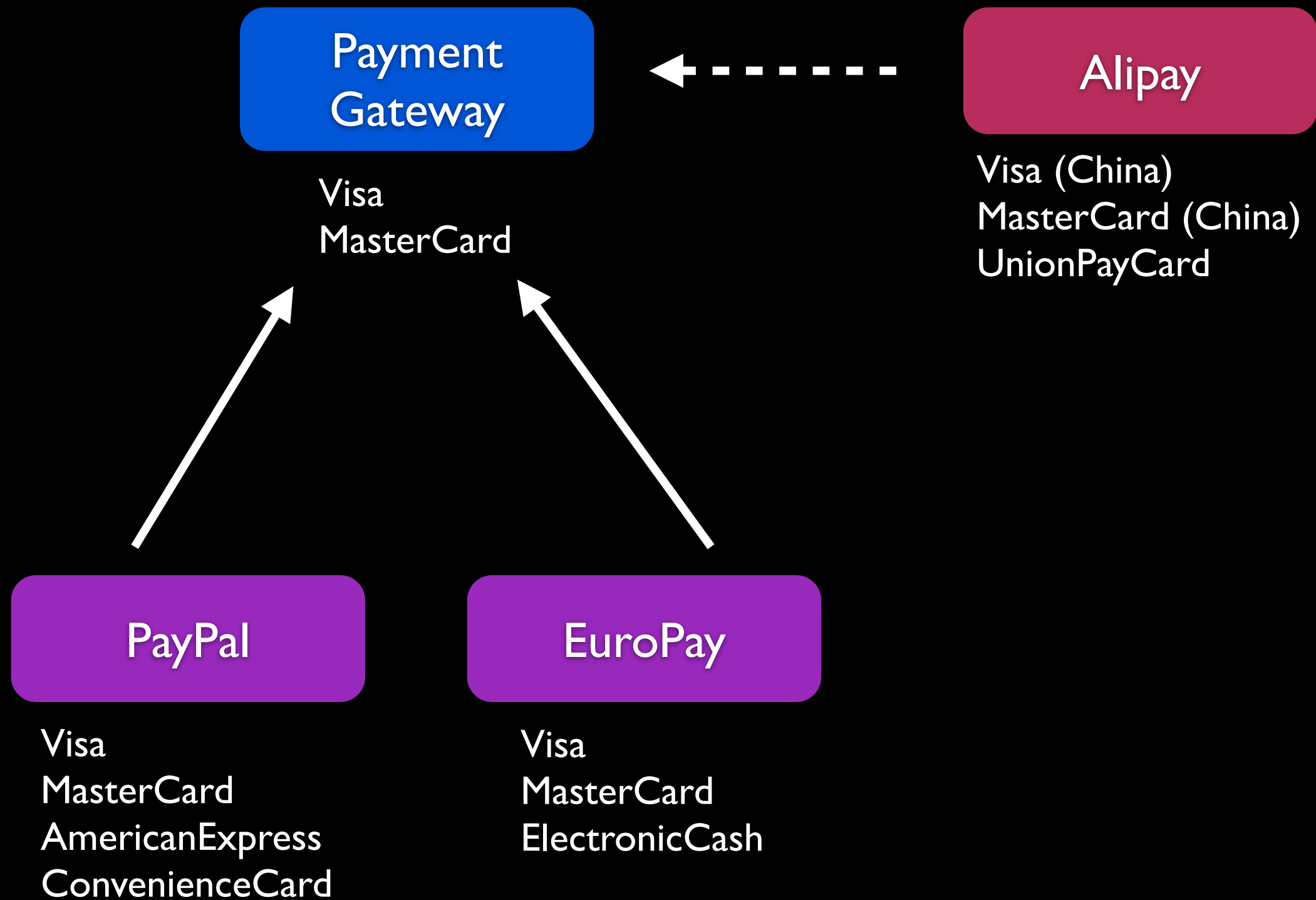
```

class PayPal extends PaymentGateway {
  val url = "http://paypal.yoda.informatik.hs-mannheim.de"
  // accept Visa and Master, and ...
  def pay(item: Billable, americanExpress: AmericanExpress): PaymentResult
  def pay(item: Billable, convenienceCard: ConvenienceCard): PaymentResult
}

class EuroPay extends PaymentGateway {
  val url = "http://europay.yoda.informatik.hs-mannheim.de"
  // accept Visa and Master, and ...
  def pay(item: Billable, electronicCash: ElectronicCash): PaymentResult
}

class Alipay {
  val url = "http://alipay.yoda.informatik.hs-mannheim.de"
  def pay(item: Billable, visa: Visa): PaymentResult = {
    // same pre-condition as PaymentGateway, plus the following one more rule
    assert( visa.issuer is a Chinese bank )
    // only accept Visa card issued by a Chinese bank
    // same post-condition as PaymentGateway
  }
  def pay(item: Billable, masterCard: MasterCard): PaymentResult {
    // same pre-condition as PaymentGateway, plus the following one more rule
    assert( masterCard.issuer is a Chinese bank )
    // only accept MasterCard issued by a Chinese bank
    // same post-condition as PaymentGateway
  }
  def pay(item: Billable, unionPayCard: UnionPayCard): PaymentResult
}

```



# Questions?

email [me@riobard.com](mailto:me@riobard.com)

visit <http://groups.google.com/group/scala-course-project>