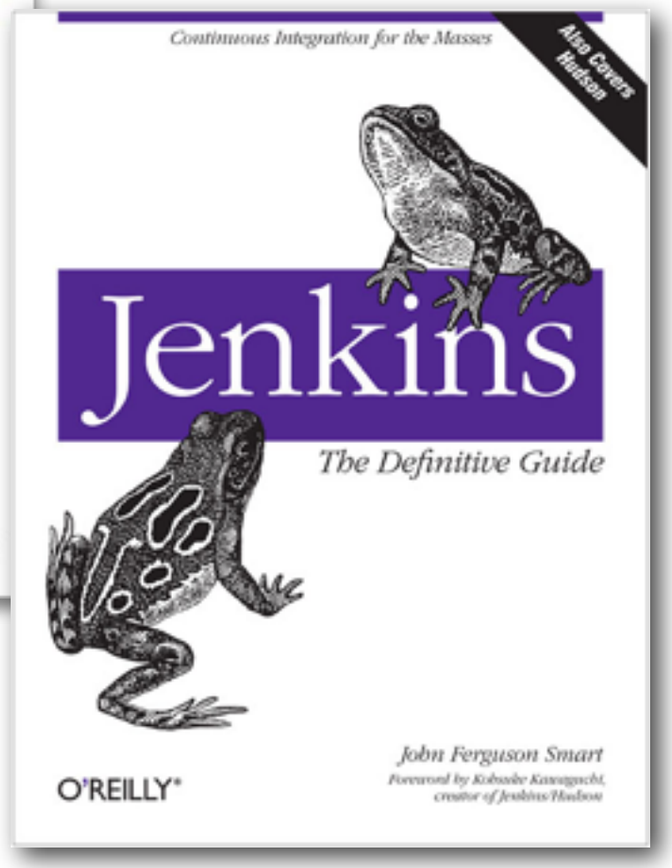
Domain Driven Design

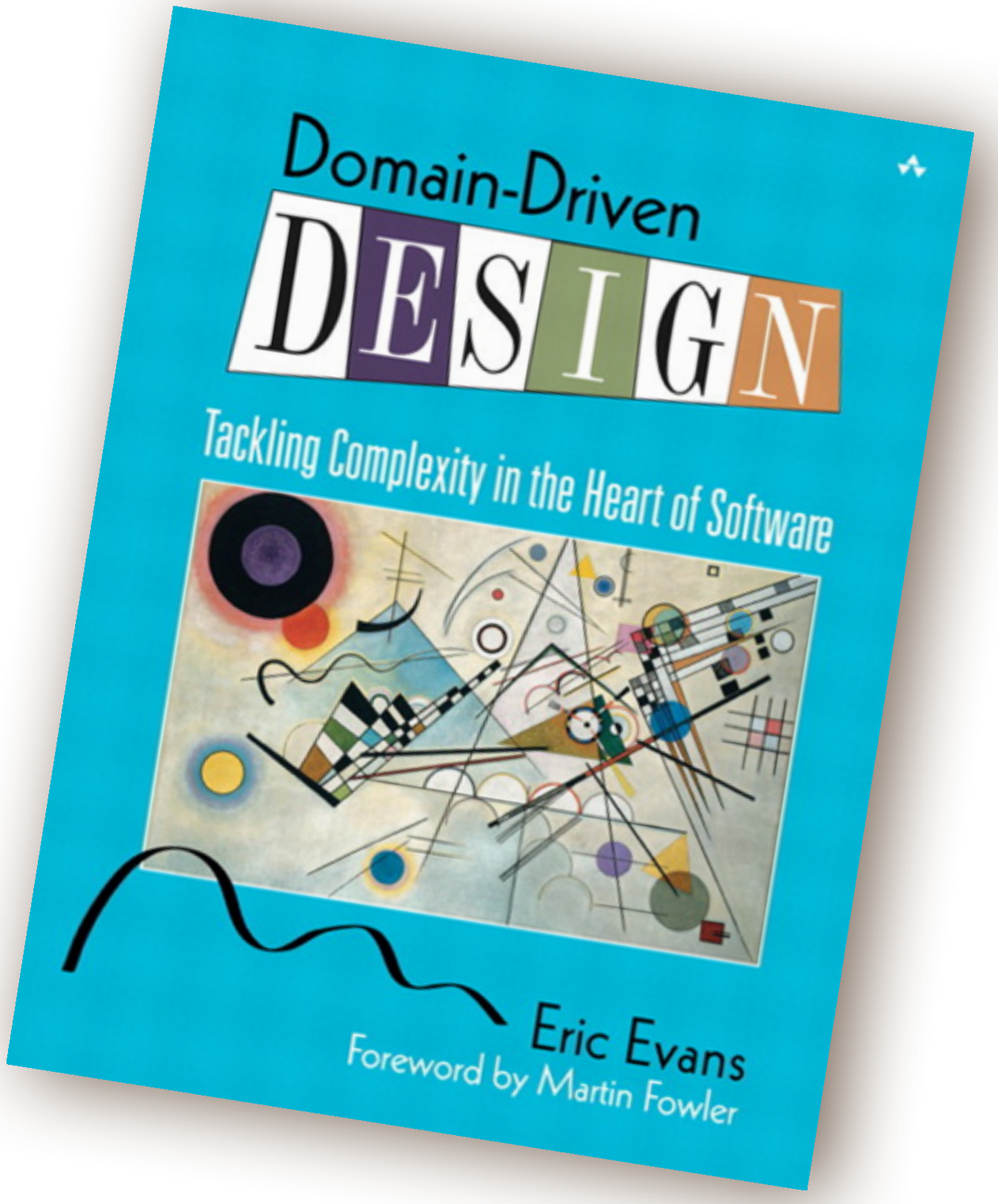
software that reflects the business domain

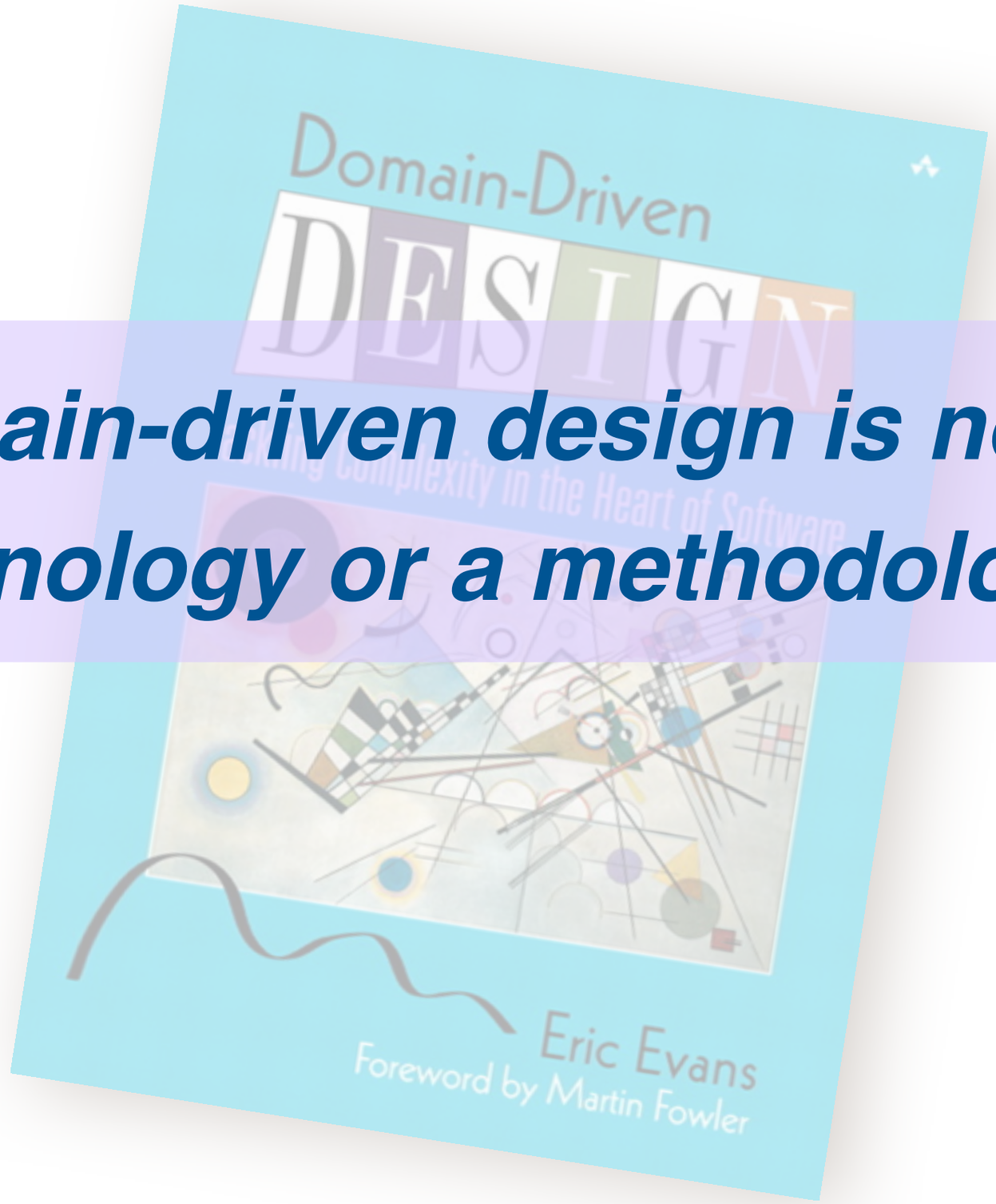
John Ferguson Smart

So who is this guy, anyway?



What is Domain-Driven Design?

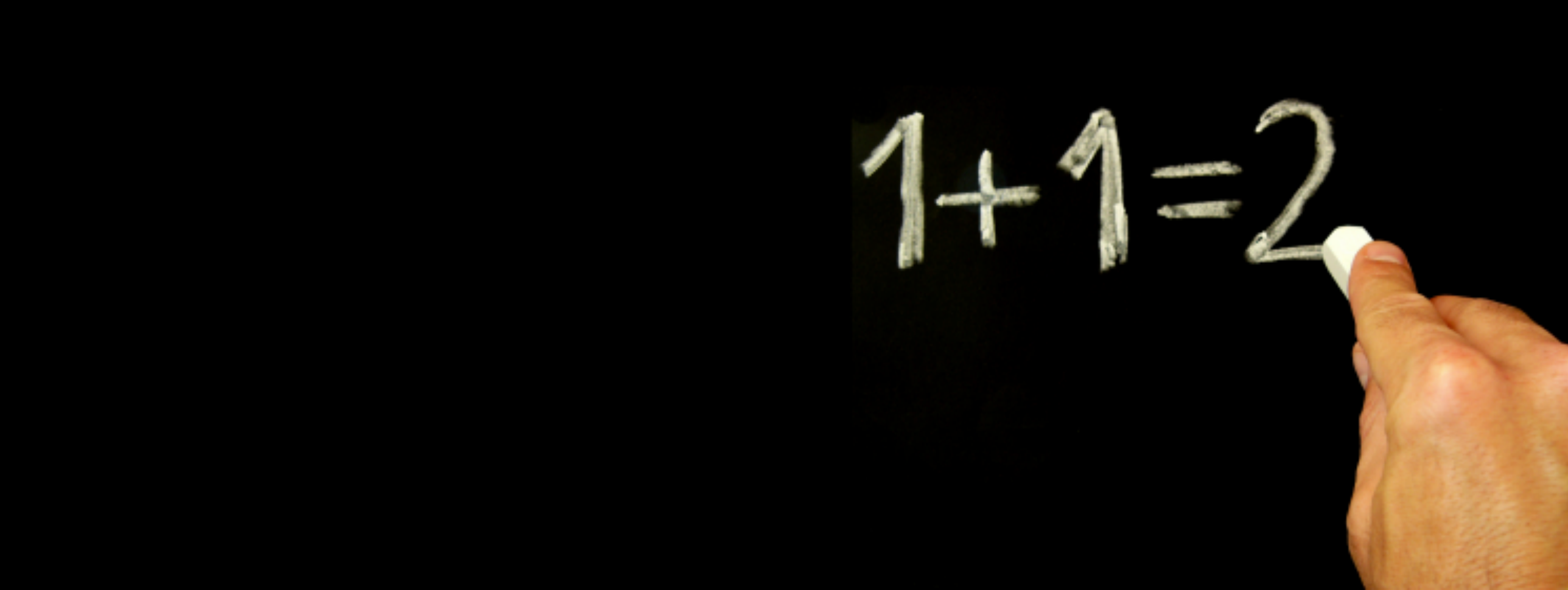




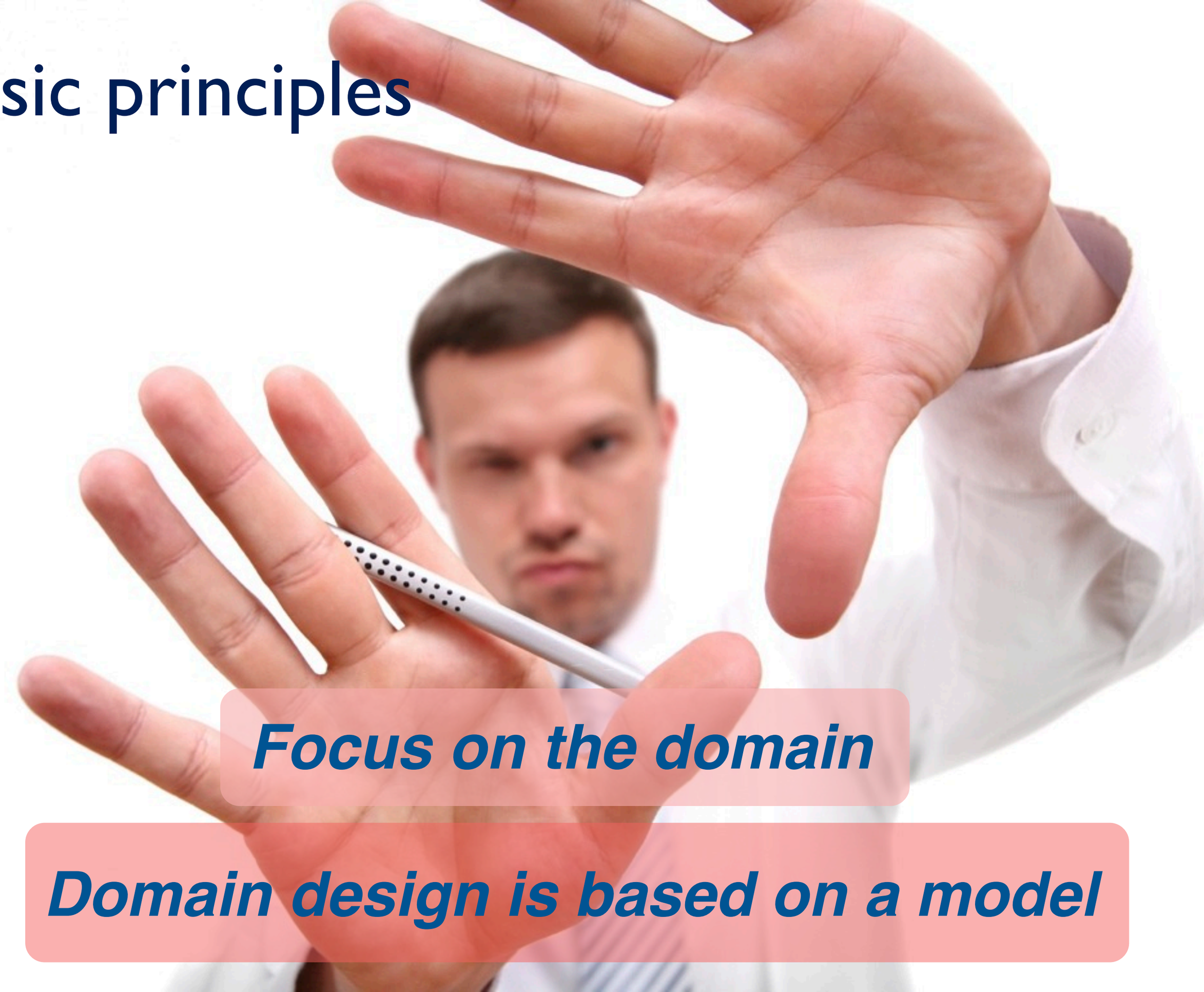
Domain-driven design is not a technology or a methodology.

It's a set of principles and patterns for focusing design effort where it matters most.

Basic principles



Basic principles



Focus on the domain

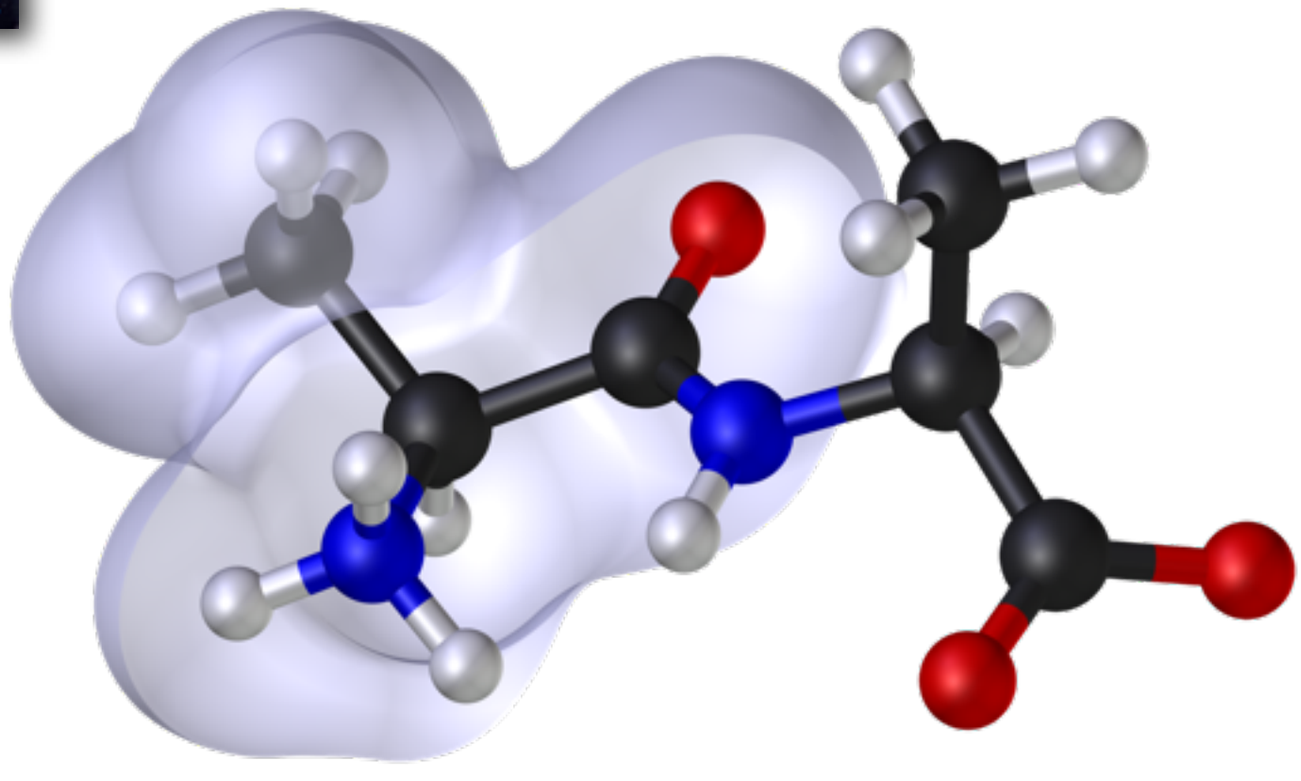
Domain design is based on a model

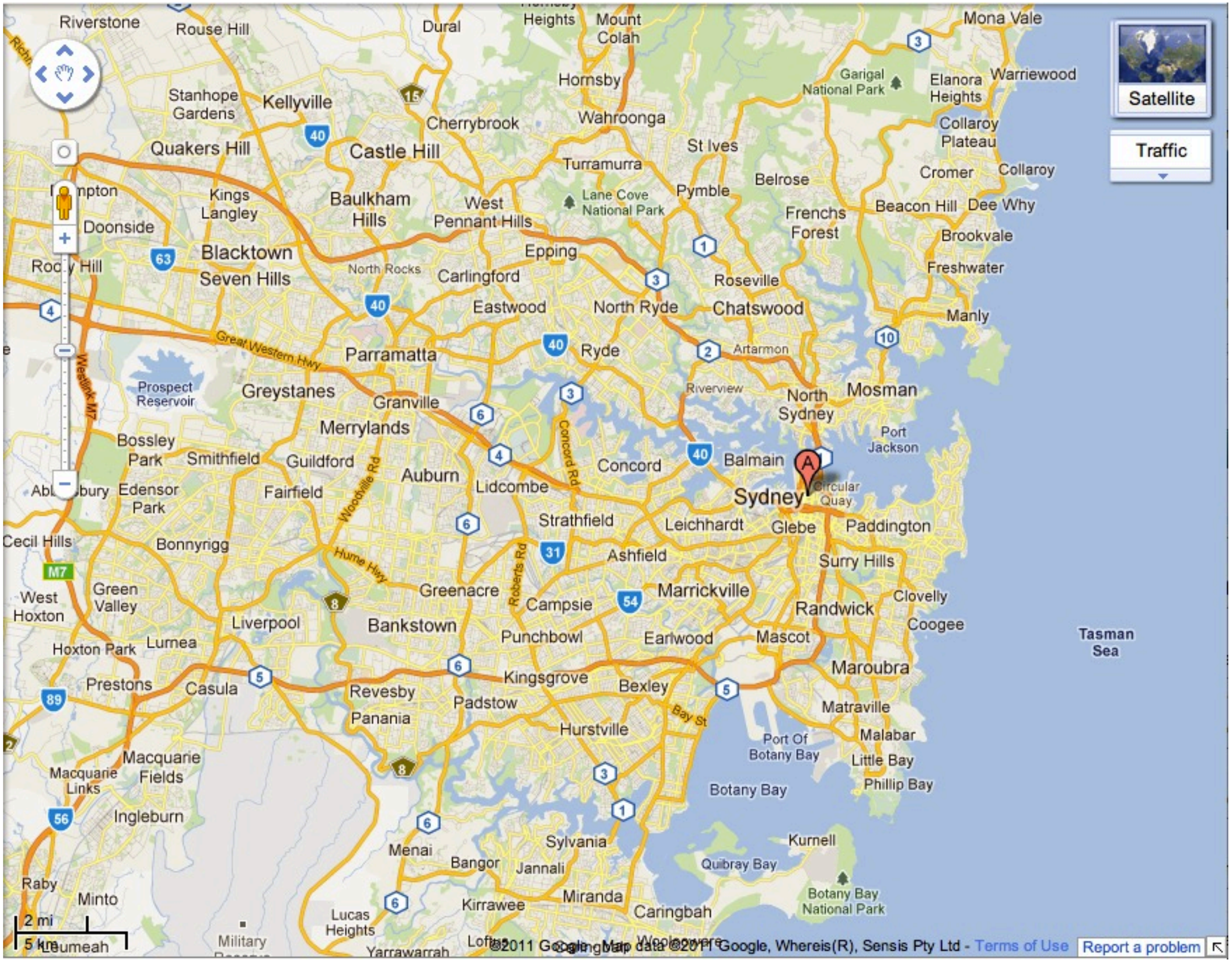
Basic principles

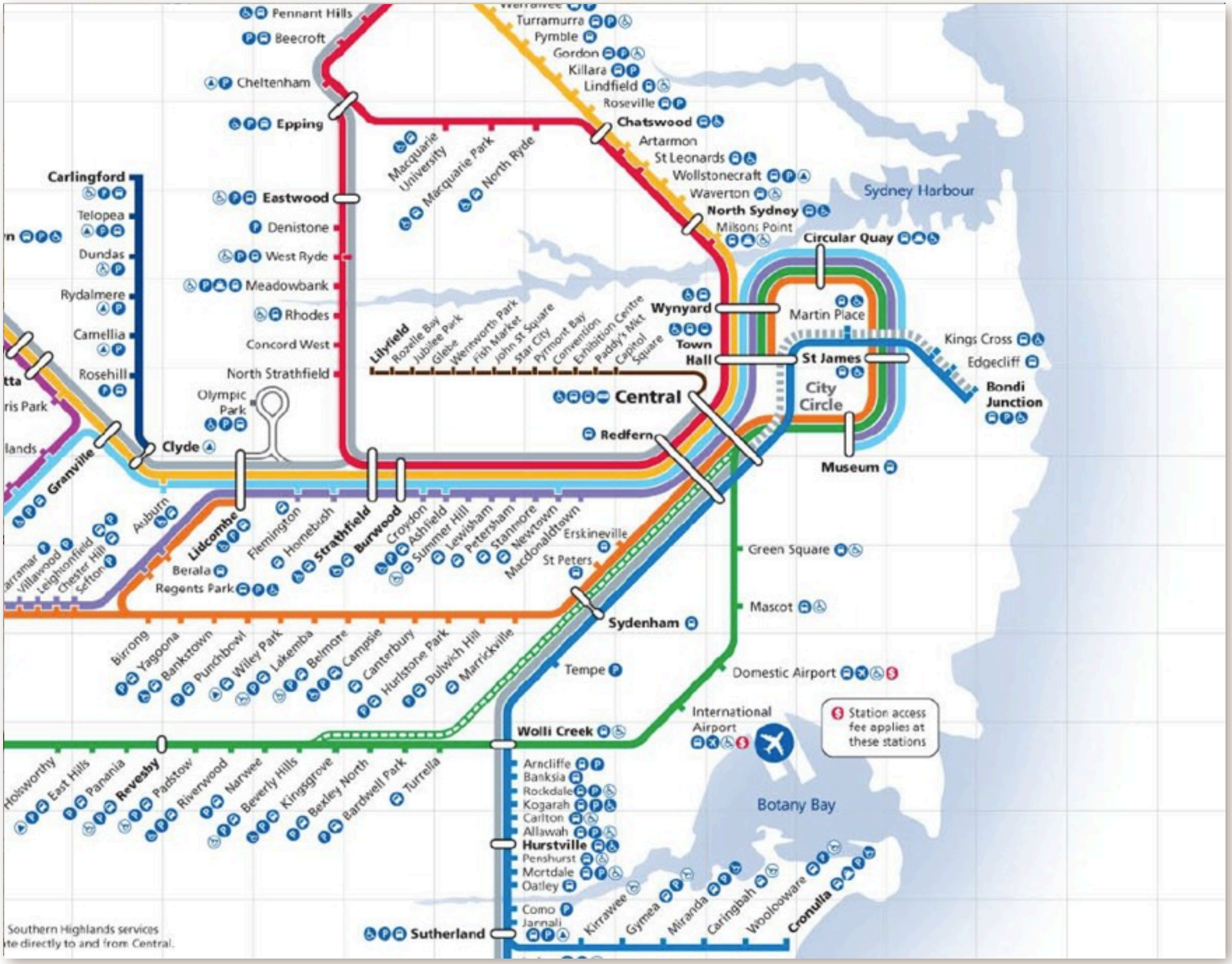
A Ubiquitous Language...

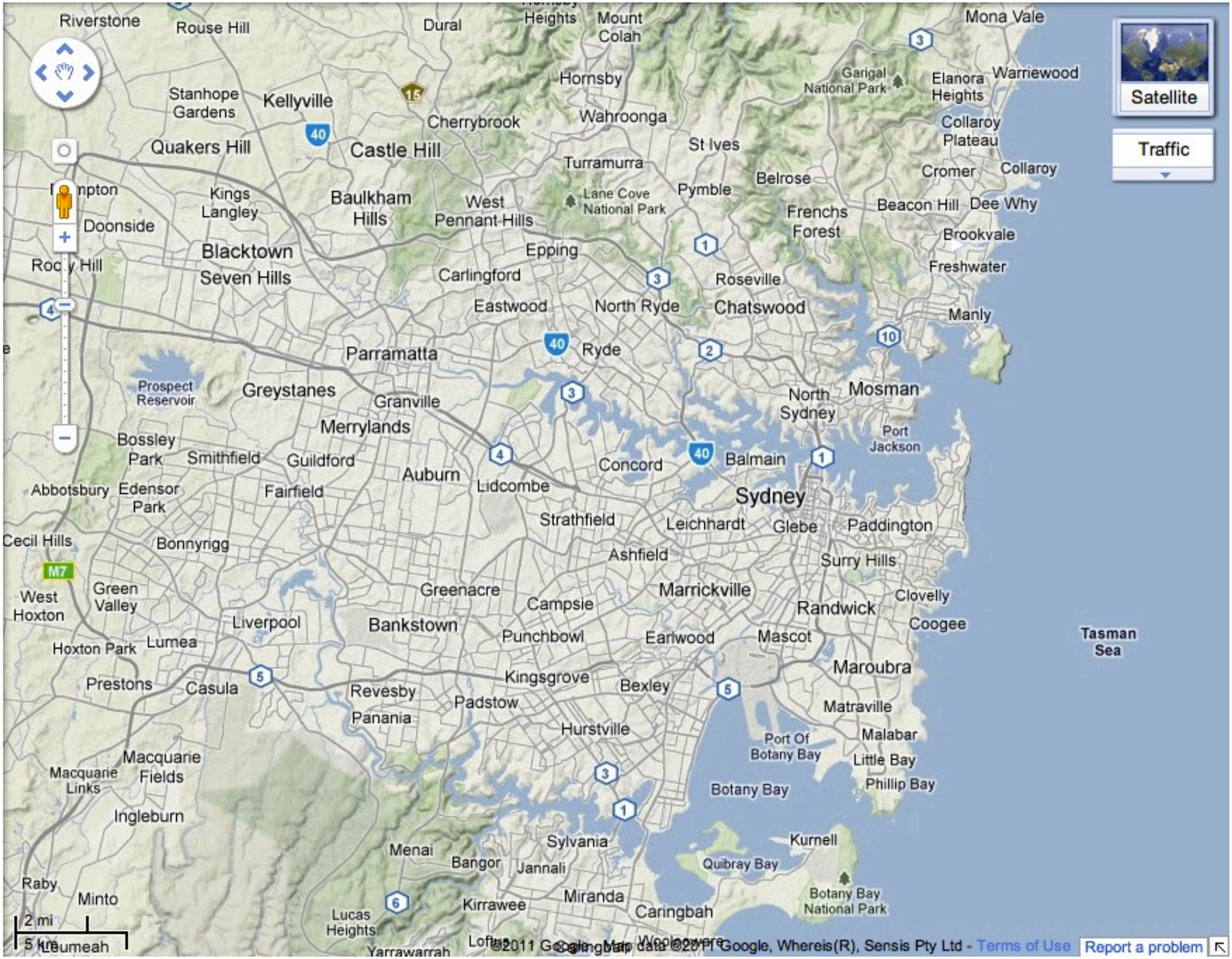
...and a shared vision

So what's a model?

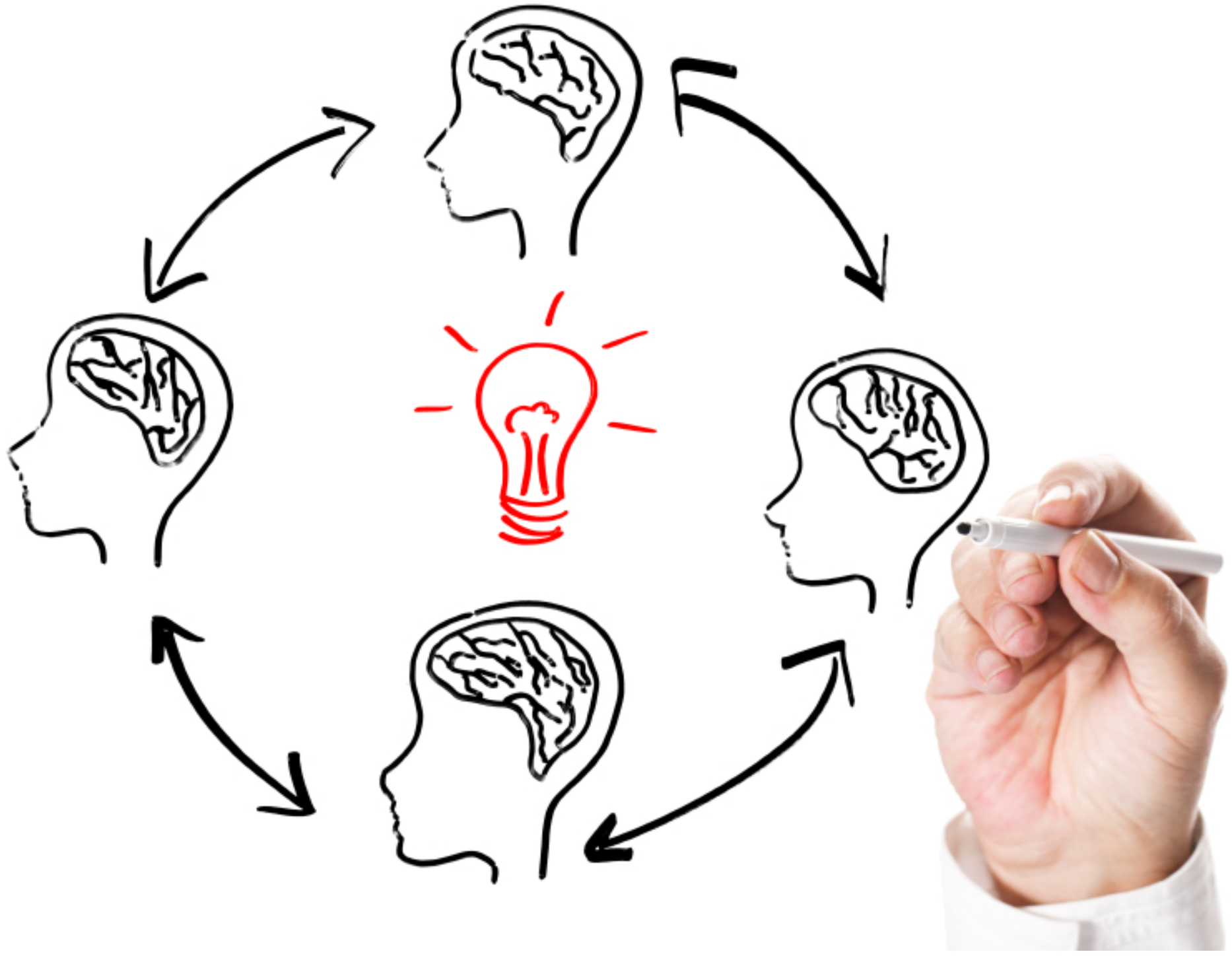



Satellite
Traffic
Riverstone
Rouse Hill
Dural
Mount Colah
Mona Vale
Hornsby
Garigal National Park
Elanora Heights
Warriewood
Stanhope Gardens
Kellyville
Cherrybrook
Wahroonga
Collaroy Plateau
Quakers Hill
Castle Hill
St Ives
Turramurra
Cromer
Collaroy
Kings Langley
Baulkham Hills
West Pennant Hills
Lane Cove National Park
Pymble
Belrose
Beacon Hill
Dee Why
Frenchs Forest
Doonside
Blacktown
Seven Hills
Epping
Brookvale
Freshwater
North Rocks
Carlingford
Roseville
Eastwood
North Ryde
Chatswood
Manly
Great Western Hwy
Parramatta
Ryde
Artarmon
Prospect Reservoir
Greystanes
Granville
Riverview
North Sydney
Mosman
Merrylands
Bossley Park
Smithfield
Guildford
Auburn
Concord
Balmain
Port Jackson
Edensor Park
Fairfield
Lidcombe
Sydney
Circular Quay
Strathfield
Leichhardt
Glebe
Paddington
Cecil Hills
Bonnyrigg
Ashfield
Surry Hills
Hume Hwy
Greenacre
Campsie
Marrickville
Randwick
Clovelly
West Hoxton
Green Valley
Liverpool
Bankstown
Punchbowl
Earlwood
Mascot
Coogee
Tasman Sea
Hoxton Park
Lurnea
Maroubra
Prestons
Casula
Revesby
Kingsgrove
Bexley
Padstow
Matraville
Panania
Hurstville
Malabar
Port Of Botany Bay
Little Bay
Macquarie Fields
Macquarie Links
Botany Bay
Phillip Bay
Ingleburn
Sylvania
Kurnell
Menai
Bangor
Jannali
Quibray Bay
Raby
Minto
Kirrawee
Miranda
Caringbah
Botany Bay National Park
Lucas Heights
Leumeah
Military
2 mi
5 km
©2011 Google - Map data ©2011 Google, Whereis(R), Sensis Pty Ltd - Terms of Use
Report a problem

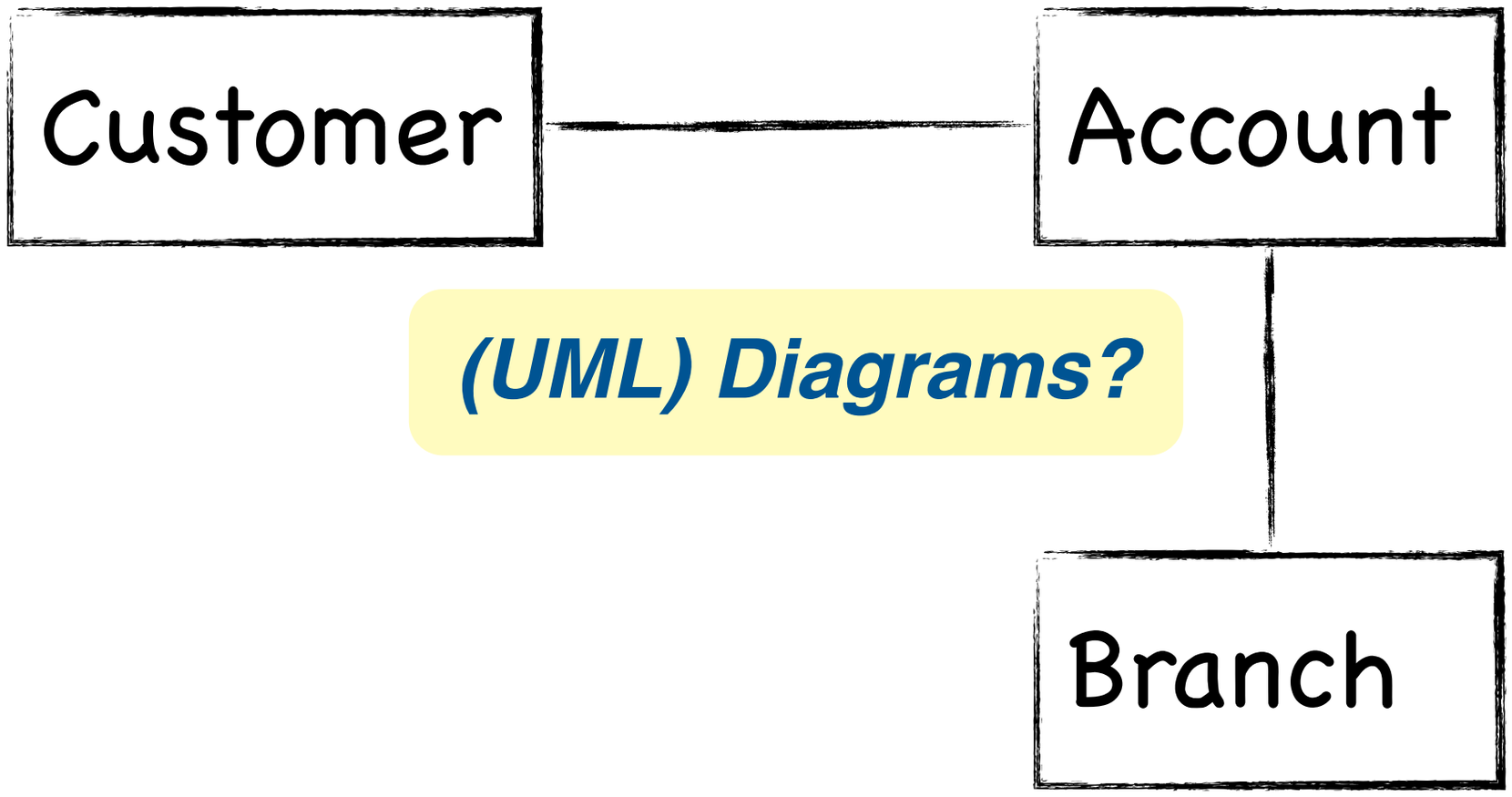

Pennant Hills
Beecroft
Cheltenham
Epping
Carlingford
Telopea
Dundas
Rydalmere
Camellia
Rosehill
Clyde
Granville
Auburn
Lidcombe
Berala
Regents Park
Flemington
Homebush
Strathfield
Burwood
Croydon
Ashfield
Summer Hill
Lewisham
Petersham
Stanmore
Newtown
Macdonaldtown
Erskineville
St Peters
Sydenham
Tempe
Macquarie University
Macquarie Park
North Ryde
Eastwood
Denistone
West Ryde
Meadowbank
Rhodes
Concord West
North Strathfield
Olympic Park
Lilyfield
Rozelle Bay
Jubilee Park
Glebe
Wentworth Park
Fish Market
John St Square
Star City
Pyrmont Bay
Convention
Exhibition Centre
Paddy's Mkt
Capitol Square
Central
Redfern
Wynyard
Town Hall
Martin Place
Circular Quay
North Sydney
Milsons Point
Chatswood
Artarmon
St Leonards
Wollstonecraft
Waverton
Turramurra
Pymble
Gordon
Killara
Lindfield
Roseville
Sydney Harbour
St James
City Circle
Museum
Kings Cross
Edgecliff
Bondi Junction
Green Square
Mascot
Domestic Airport
International Airport
Wolli Creek
Arncliffe
Banksia
Rockdale
Kogarah
Carlton
Allawah
Hurstville
Penshurst
Mortdale
Oatley
Como
Jannali
Sutherland
Kirrawee
Gymea
Miranda
Caringbah
Wooloowara
Cronulla
Botany Bay
Station access fee applies at these stations
Birrong
Yagoona
Bankstown
Punchbowl
Wiley Park
Lakemba
Belmore
Campsie
Canterbury
Hurlstone Park
Dulwich Hill
Marrickville
Holsworthy
East Hills
Panania
Revesby
Padstow
Riverwood
Narwee
Beverly Hills
Kingsgrove
Bexley North
Bardwell Park
Turrella
Southern Highlands services
te directly to and from Central.


Riverstone
Rouse Hill
Dural
Hornsby Heights
Mount Colah
Mona Vale
Garigal National Park
Elanora Heights
Warriewood
Satellite
Traffic
Stanhope Gardens
Kellyville
Cherrybrook
Hornsby
Wahroonga
Collaroy Plateau
Quakers Hill
Castle Hill
St Ives
Turramurra
Cromer
Collaroy
Kings Langley
Baulkham Hills
West Pennant Hills
Lane Cove National Park
Pymble
Belrose
Beacon Hill
Dee Why
Doonside
Frenchs Forest
Brookvale
Blacktown
Seven Hills
Epping
Freshwater
Carlingford
Roseville
Eastwood
North Ryde
Chatswood
Manly
Parramatta
Ryde
Prospect Reservoir
Greystanes
Granville
North Sydney
Mosman
Merrylands
Port Jackson
Bossley Park
Smithfield
Guildford
Auburn
Concord
Balmain
Abbotsbury
Edensor Park
Fairfield
Lidcombe
Sydney
Strathfield
Leichhardt
Glebe
Paddington
Cecil Hills
Bonnyrigg
Ashfield
Surry Hills
West Hoxton
Green Valley
Greenacre
Campsie
Marrickville
Randwick
Clovelly
Liverpool
Bankstown
Punchbowl
Earlwood
Mascot
Coogee
Hoxton Park
Lurnea
Tasman Sea
Prestons
Casula
Revesby
Kingsgrove
Bexley
Maroubra
Padstow
Matraville
Panania
Hurstville
Port Of Botany Bay
Malabar
Macquarie Fields
Little Bay
Macquarie Links
Botany Bay
Phillip Bay
Ingleburn
Sylvania
Kurnell
Menai
Bangor
Jannali
Quibray Bay
Raby
Minto
Miranda
Botany Bay National Park
Kirrawee
Caringbah
Lucas Heights
Leumeah
Yarrawarrah
2 mi
5 km
©2011 Google - Map data ©2011 Google, Whereis(R), Sensis Pty Ltd - Terms of Use
Report a problem

How do you represent your domain model?



How do you represent your domain model?



(UML) Diagrams?

How do you represent your domain model?

Detailed specifications?

How do you represent your domain model?

Account Holder withdraws cash
As an *account holder*
I want to *withdraw cash* from an *ATM*
so that I can get money out when the bank is closed

Scenario 1: *Account* has *sufficient* funds
Given the account balance is $100
And the *card is valid*
And the *machine contains enough money*
When the Account Holder requests $20
Then the ATM should dispense $20
And the account balance should be $80
And the card should be returned

Scenario 2: Account has insufficient funds
Given the account balance is $10
And the card is valid
And the machine contains enough money
When the Account Holder requests $20
The ATM should not dispense any money

Free text?

How do you represent your domain model?

```
@Test
public void aCashWithdrawalShouldDeductSumFromBalance() {
    Account account = new Account();
    account.makeDeposit(100);
    account.makeCashWithdraw(60);
    assertThat(account.getBalance(), is(40));
}
```

Automated tests?

```
import com.wakaleo.bankonline.domain.Account

scenario "Make initial deposit onto a new account", {
    given "a new account",{
       account = new Account()
    }
    when "an initial deposit is made", {
       initialAmount = 100
       account.makeDeposit(initialAmount)
    }
    then "the balance should be equal to the amount deposited", {
        account.balance.shouldBe initialAmount
    }
```

How do you represent your domain model?

```
class Account(val owner : Customer,
              val accountNumber : String,
              val accountType : AccountType,
              var balance : Double) {

  def deposit(amount : Double) { balance += amount }
  def withdraw(amount : Double) { balance -= amount }
}
```

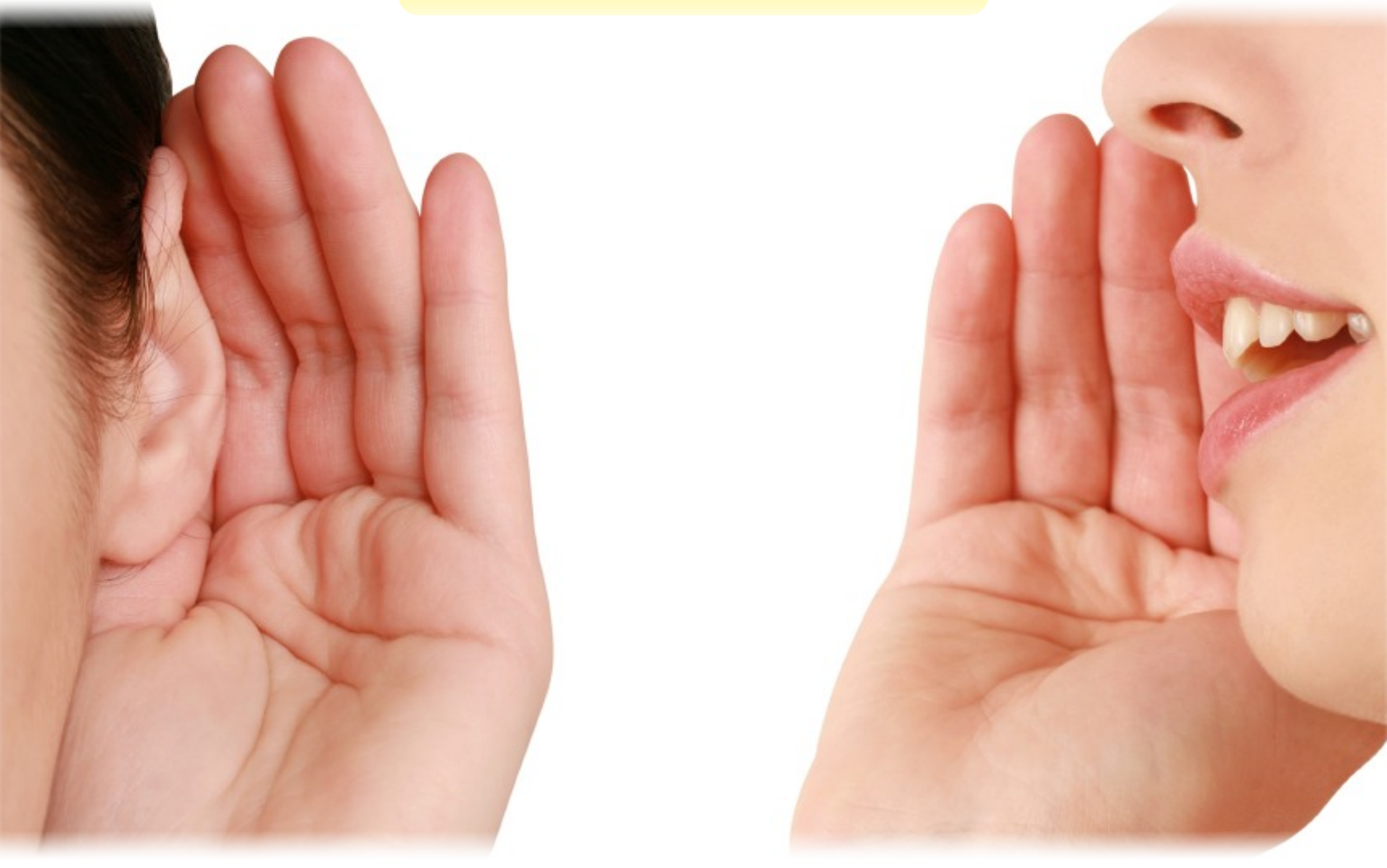
Code?

```
class Customer(val firstName : String,
               val lastName : String) {

  createAccount(accountType : AccountType) : Account
}

class AccountType extends Enumeration {
  val Current = value("CURRENT")
  val Savings = value("SAVINGS")
}
```

The model is the mental representation

Everything else is just a communication tool

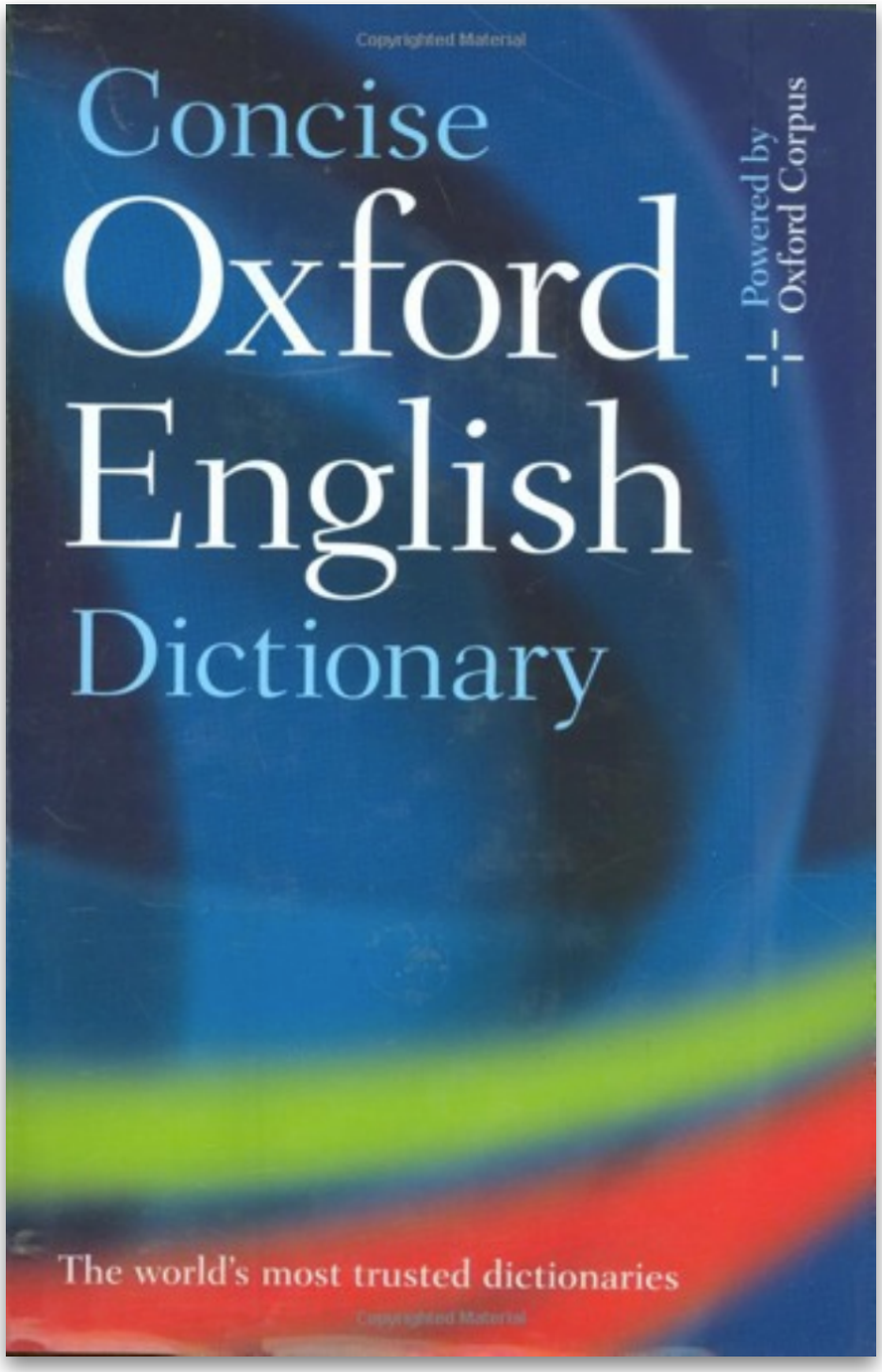
Elaborating the model



Elaborating the model

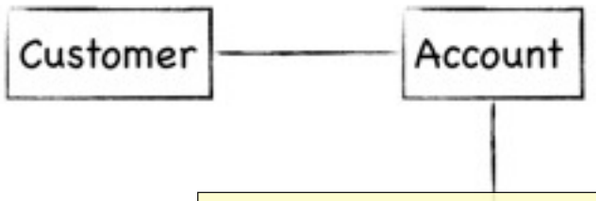
A collaborative exercise

Elaborating the model



...based on a common language

Elaborating the model



Account Holder withdraws cash
As an *account holder*
I want too *withdraw cash* from an *ATM*
so that I can get money out when the bank is closed

```
import com.wakaleo.bankonline.domain.Account

scenario "Make initial deposit onto a new account", {
    given "a new account",{
       account = new Account()
    }
    when "an initial deposit is made", {
       initialAmount = 100
       account.makeDeposit(initialAmount)
    }
    then "the balance should be equal to the amount deposited", {
        account.balance.shouldBe initialAmount
    }
```

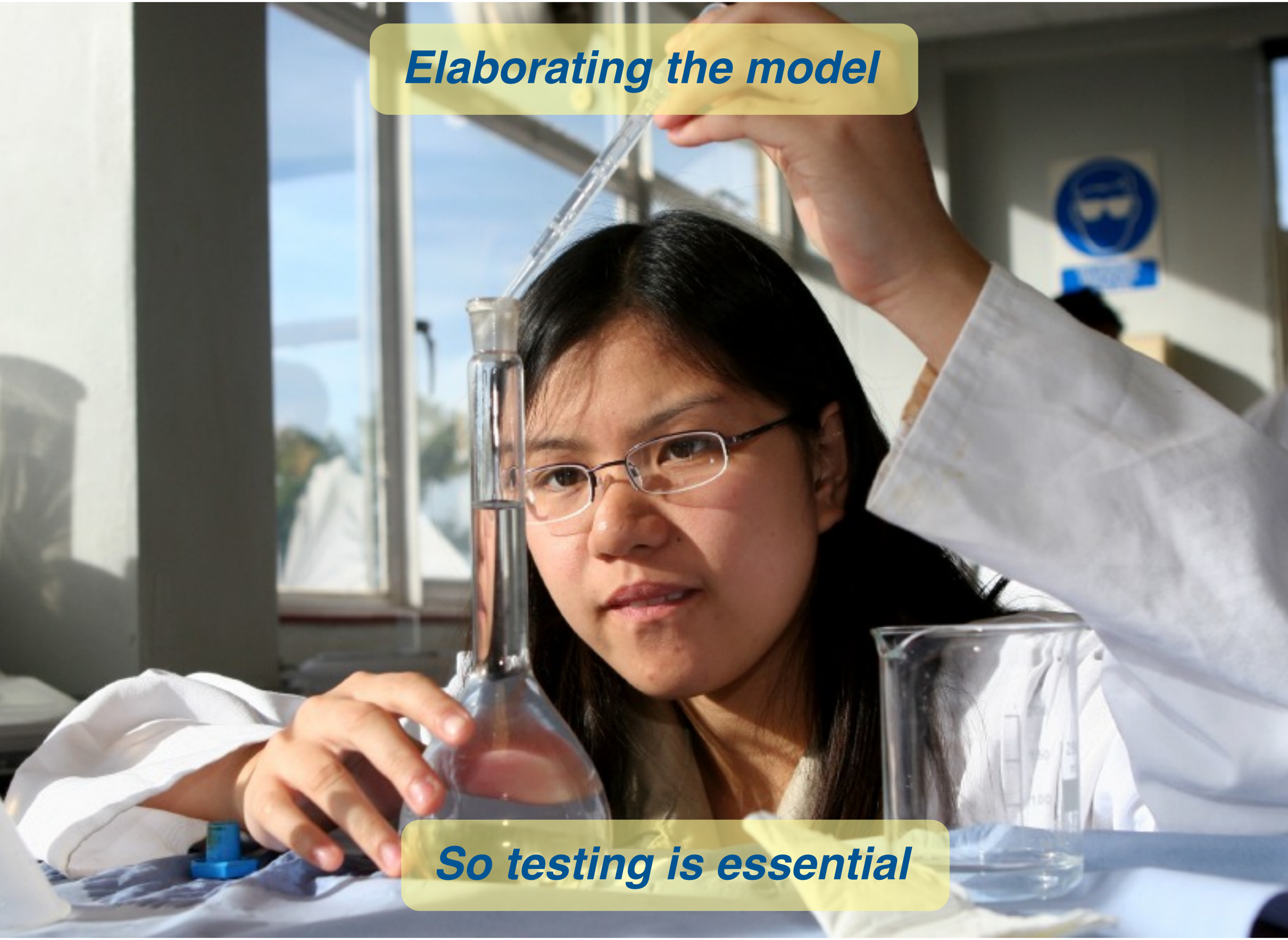
```
@Test
public void aCashWithdrawalShouldDeductSumFromBalance() {
    Account account = new Account();
    account.makeDeposit(100);
    account.makeCashWithdraw(60);
    assertThat(account.getBalance(), is(40));
}
```

```
class Account(val owner : Customer,
              val accountNumber : String,
              val accountType : AccountType,
              var balance : Double) {

  def deposit(amount : Double) { balance += amount }
  def withdraw(amount : Double) { balance -= amount }
}
```

...and expressed at all levels

Elaborating the model

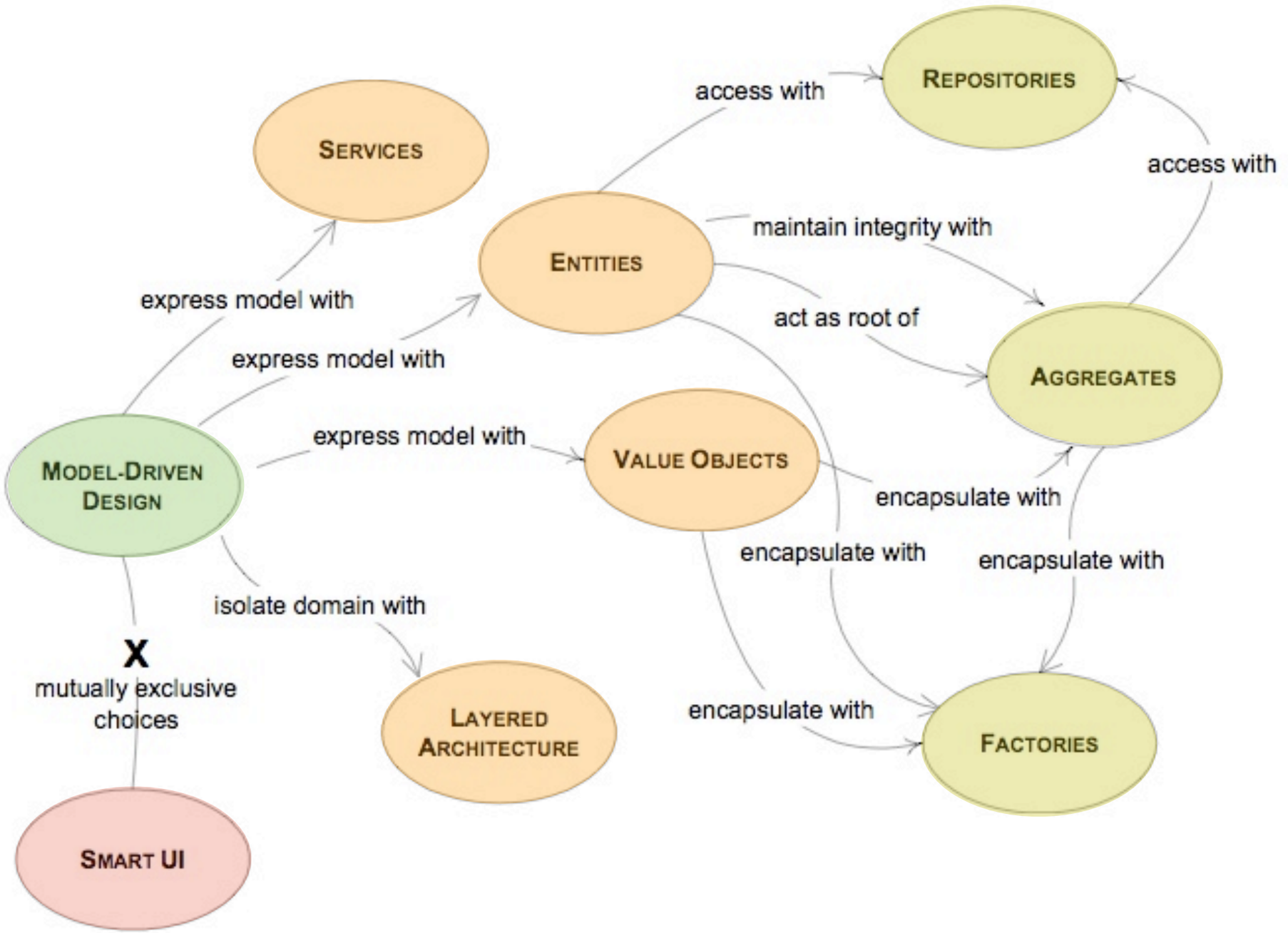
An evolutionary process


Elaborating the model
So testing is essential

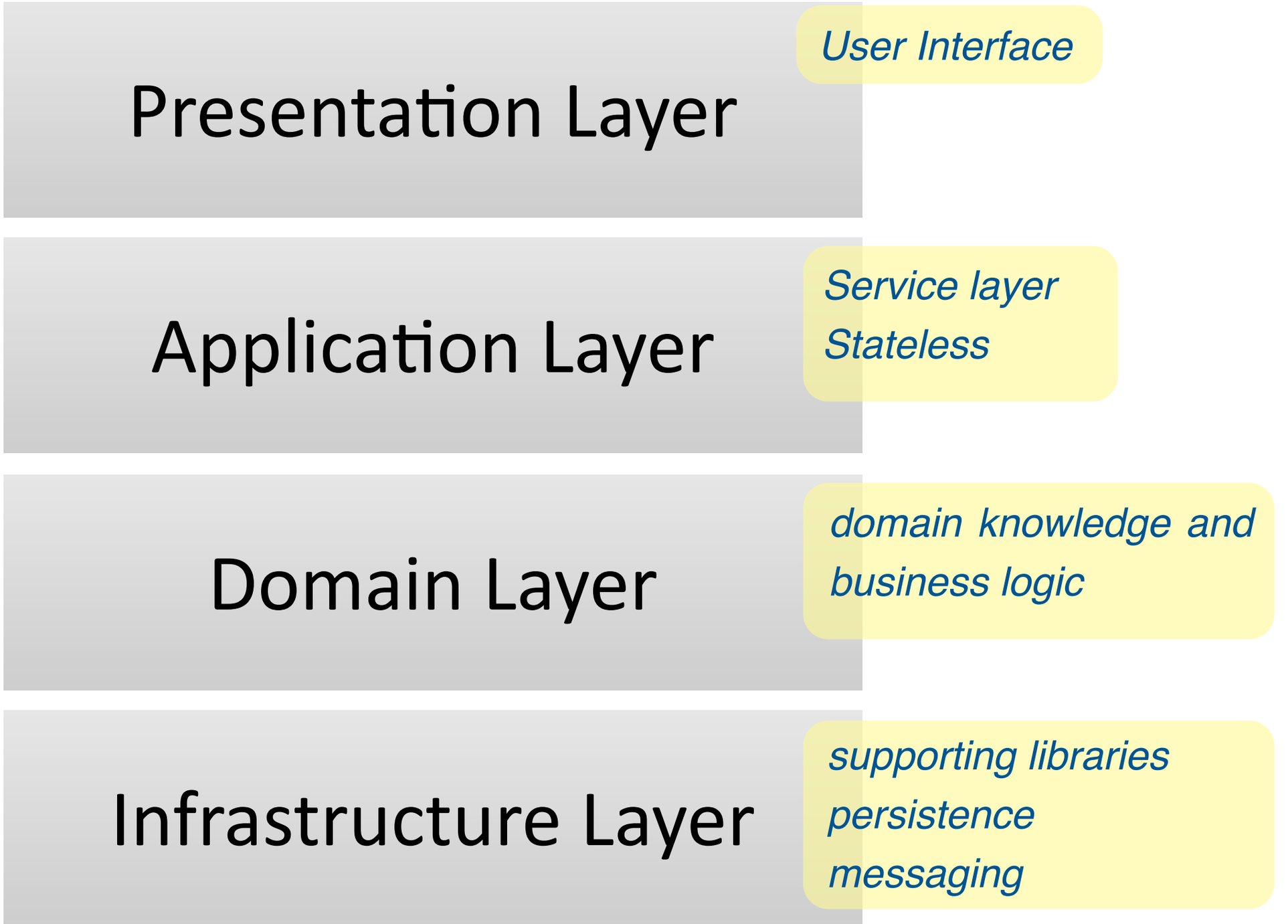
Elaborating the model

...automated testing is essential

Domain-Driven Design practices



Domain-Driven Design architecture



Domain-Driven Design architecture

Entities

- *Identify and state*
- *Persistence*

Domain-Driven Design architecture

Value Objects

- *No conceptual identity*
- *Immutable*

Domain-Driven Design architecture

Services

- *Domain-level services*
- *Stateless*

Domain-Driven Design architecture

Repositories

- *Data retrieval*
- *Abstraction of the persistence layer*

Domain-Driven Design architecture

Factories

- *Creation of complex entities*

Application code should be a reflection of the business domain

John Ferguson Smart
Email: john.smart@wakaleo.com
Web: http://www.wakaleo.com
Twitter: wakaleo