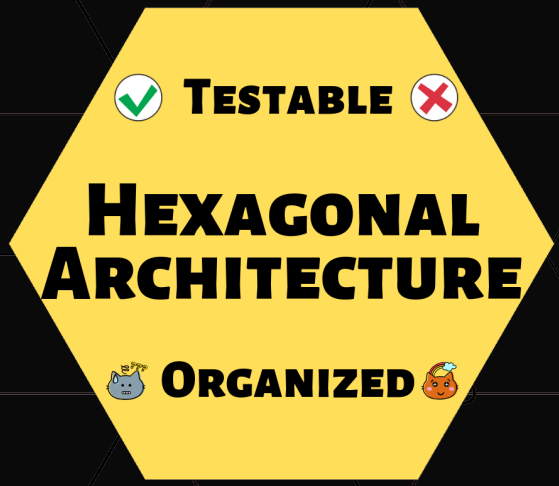
More Testable Code

With the Hexagonal Architecture

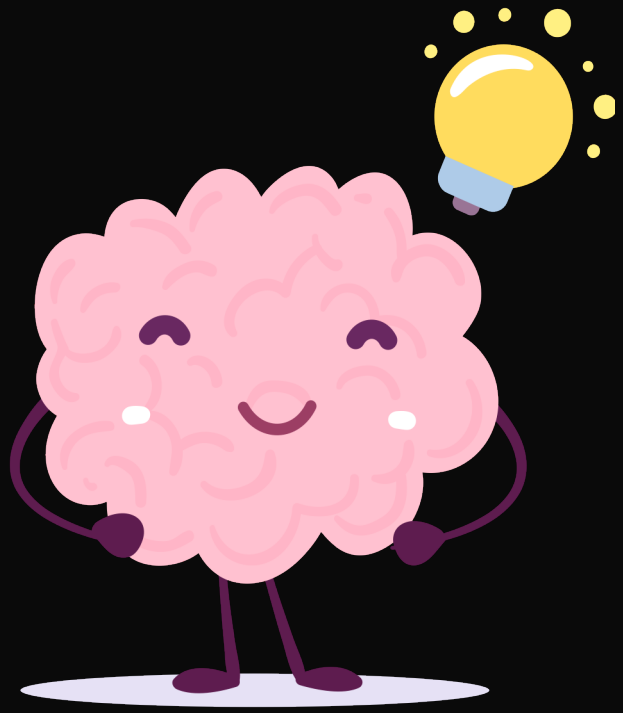
Ted M. Young
Java Trainer/Coach



Spiral Learning, LLC
ted@SpiralLearningLLC.com

@JitterTed (Twitter & Twitch)
https://ted.dev (Web)

Hold Questions to the End



...if you can.

Otherwise, ask!

Programming Language?

Ted M. Young

ted@SpiralLearningLLC.com

@JitterTed

I Can Help Your Team…

Write more Testable code with more Effective unit tests

Become better developers

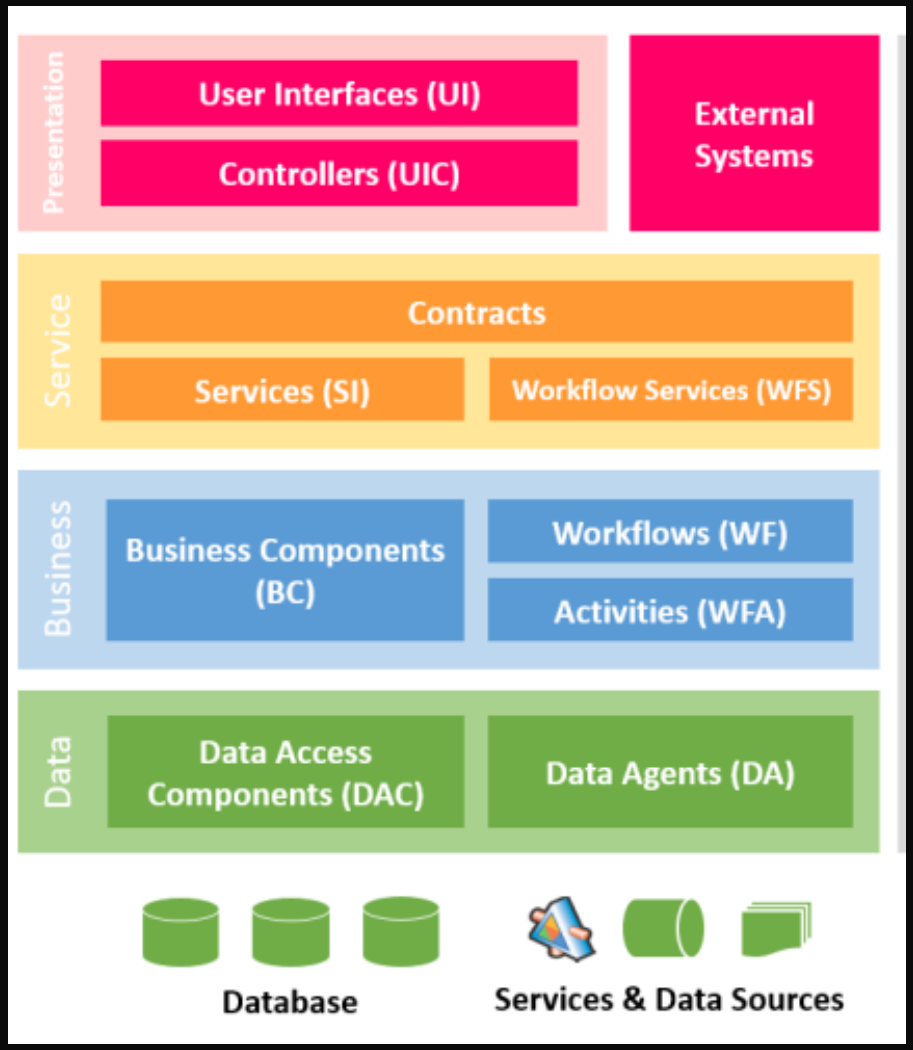
Be more productive in **Java & Spring**

Effectively use **TDD**

Apply **eXtreme Programming**

Vertically-Layered Architecture

aka
Cake Architecture



Adapted from
https://blogs.msdn.microsoft.com/malaysia/2013/08/02/layered-architecture-for-net/

Ted M. Young https://MoreTestable.com

@JITTERTED

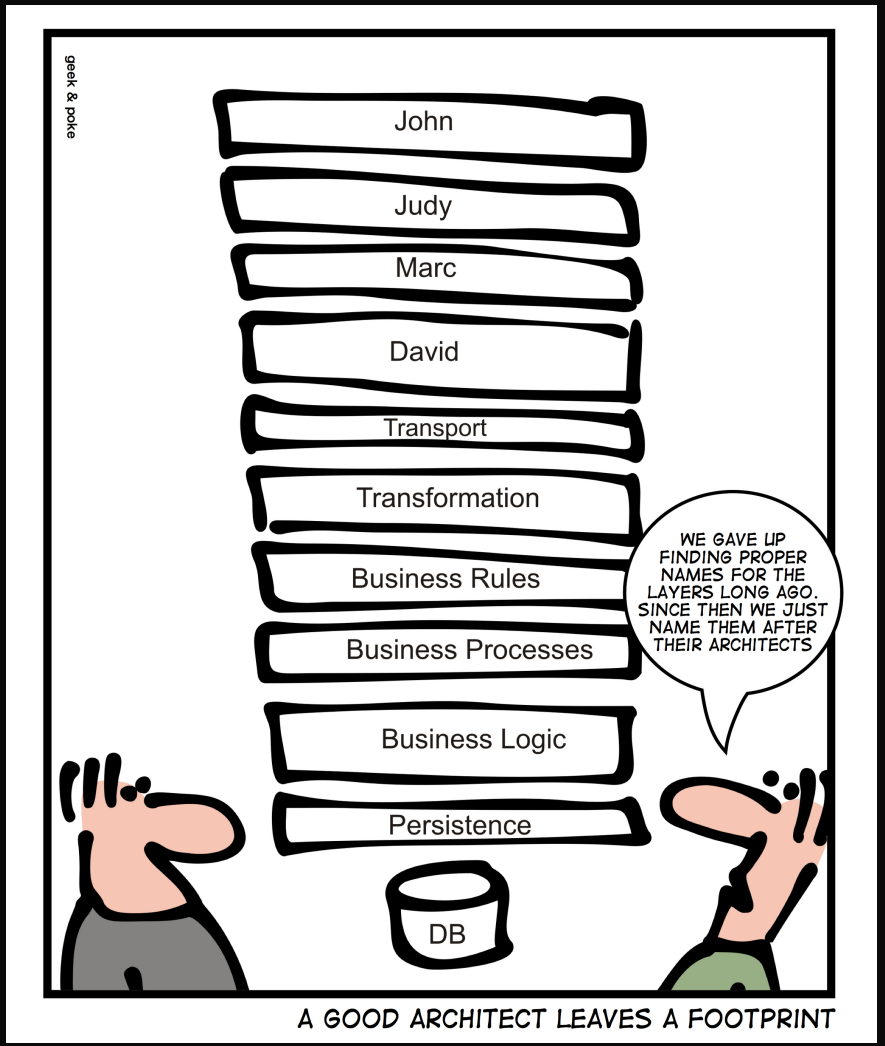
@JitterTed

Database-Centric

instead of Business-Centric

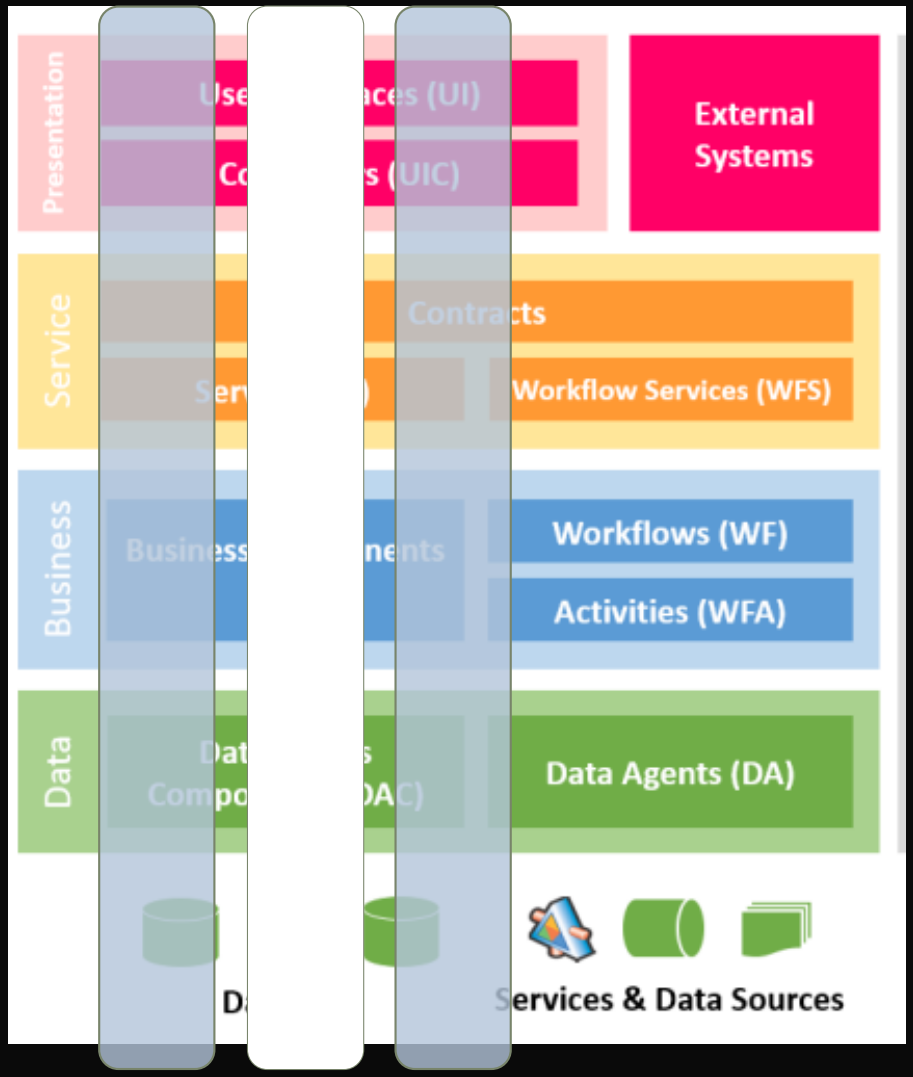
Ted M. Young https://MoreTestable.com

Vertically-Layered Architecture



"Footprints" from Geek & Poke
http://geek-and-poke.com/geekandpoke/2013/7/13/foodprints

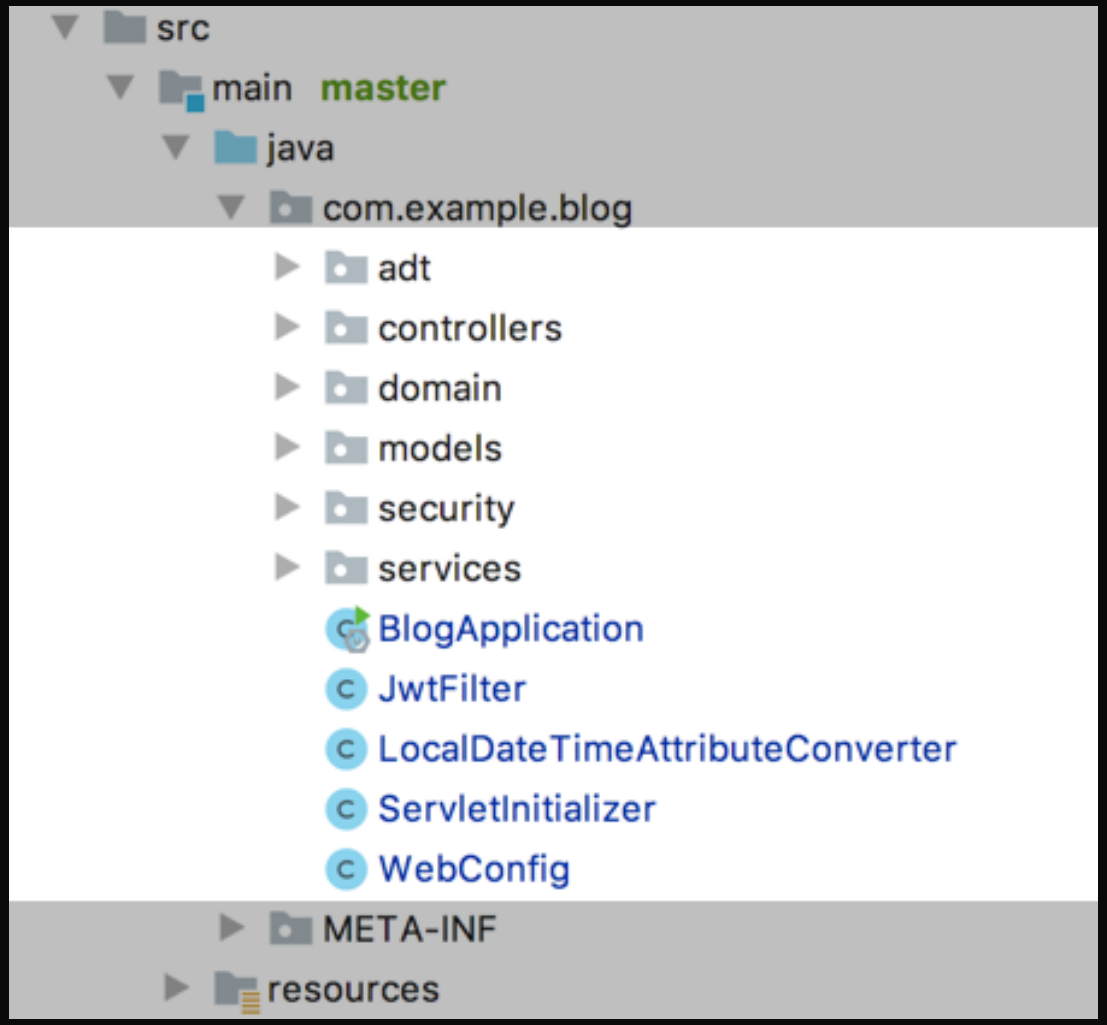
Thin Vertical Slices



@JitterTed

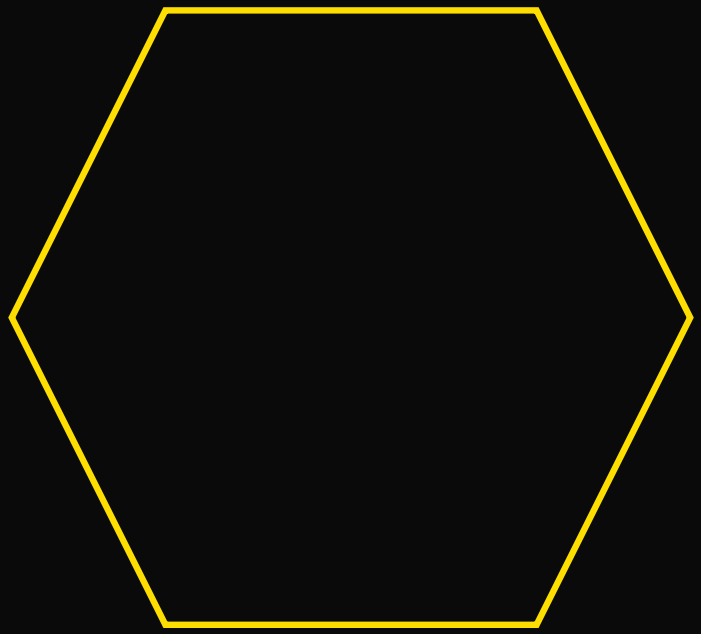
Ted M. Young https://MoreTestable.com

Vertical Package Structure




Hexagonal
Architecture

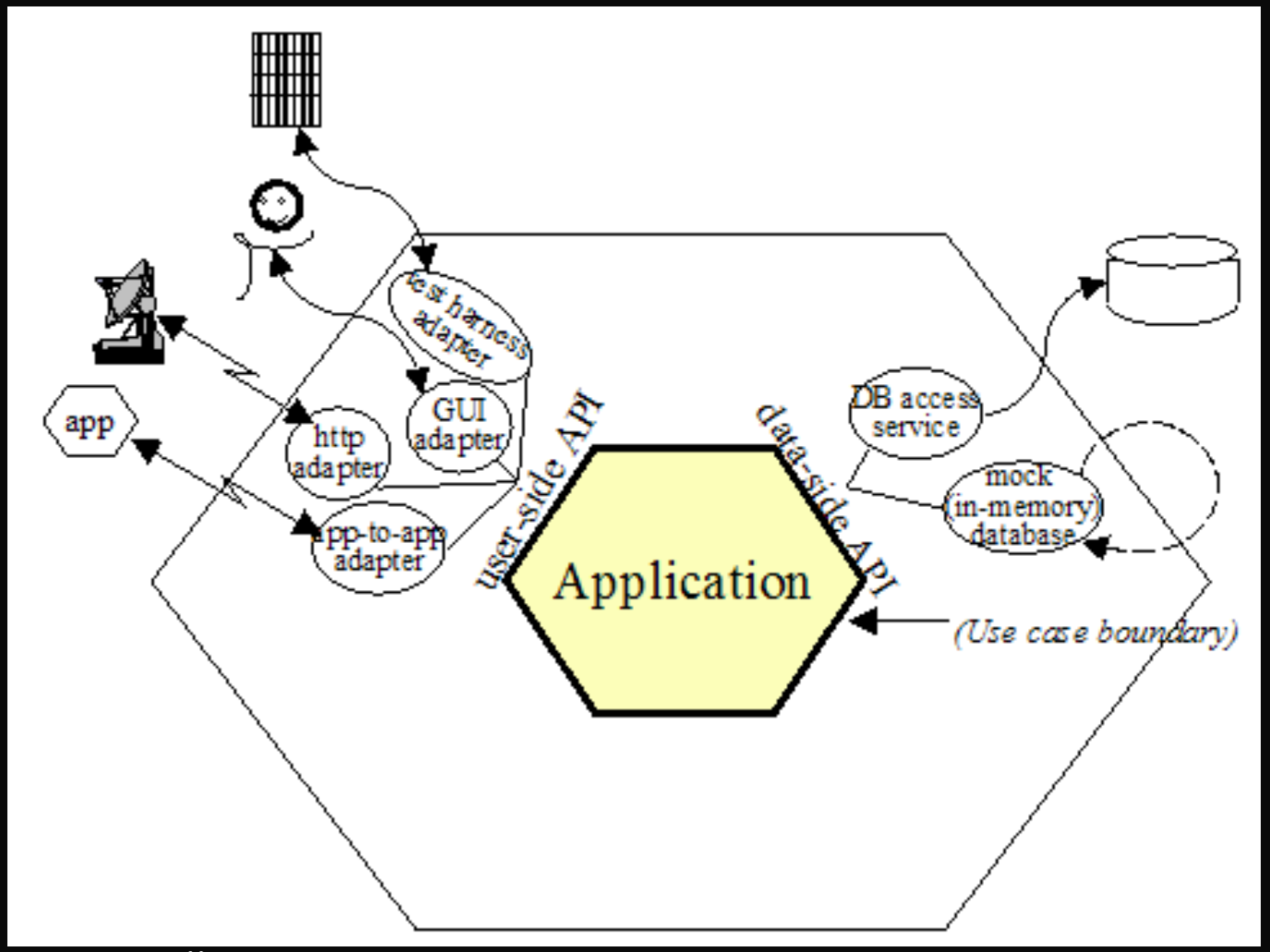
@JitterTed



Ted M. Young https://MoreTestable.com

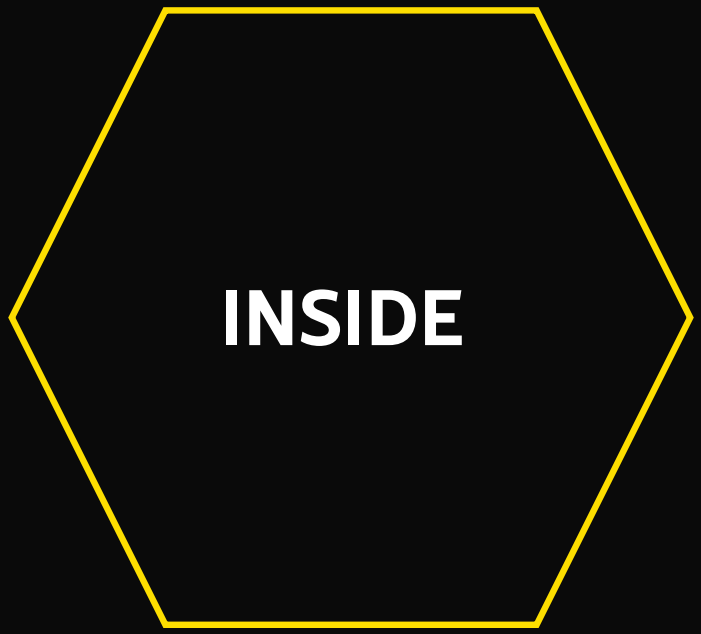
"Allow an application to equally be driven by users, programs, [and] automated tests... and to be developed and tested in isolation from its eventual run-time devices and databases."

Ports & Adapters Pattern
Alistair Cockburn


test harness adapter
GUI adapter
app
http adapter
app-to-app adapter
user-side API
data-side API
Application
DB access service
mock (in-memory) database
(Use case boundary)

OUTSIDE

OUTSIDE

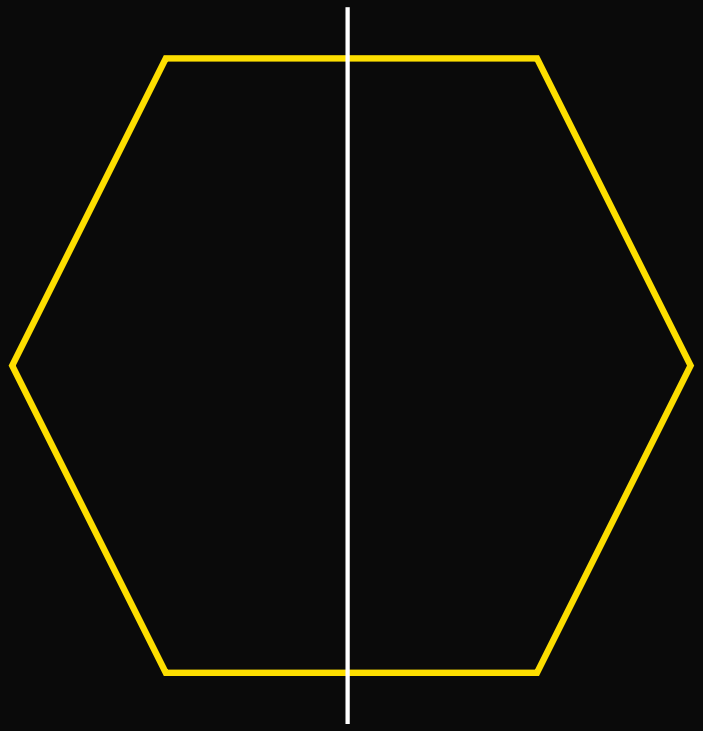


OUTSIDE

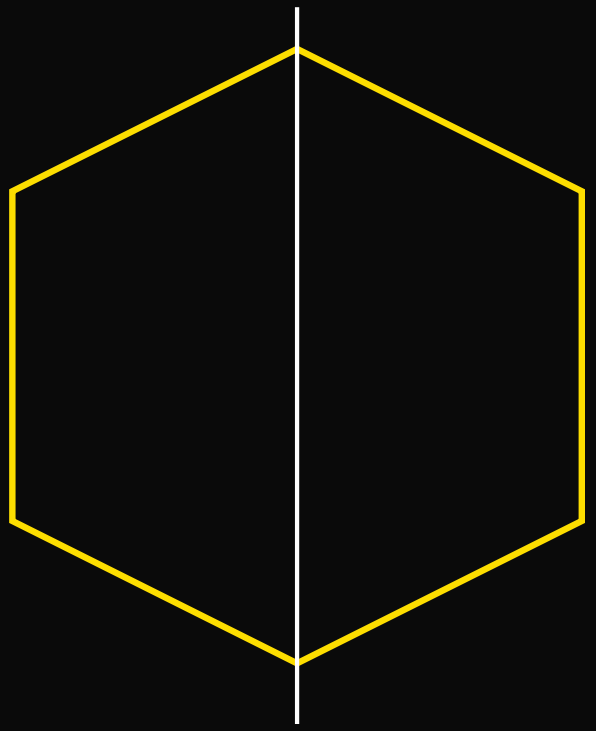
OUTSIDE

@JitterTed

FLAT-TOPPED



POINTY-TOPPED



Ted M. Young https://MoreTestable.com

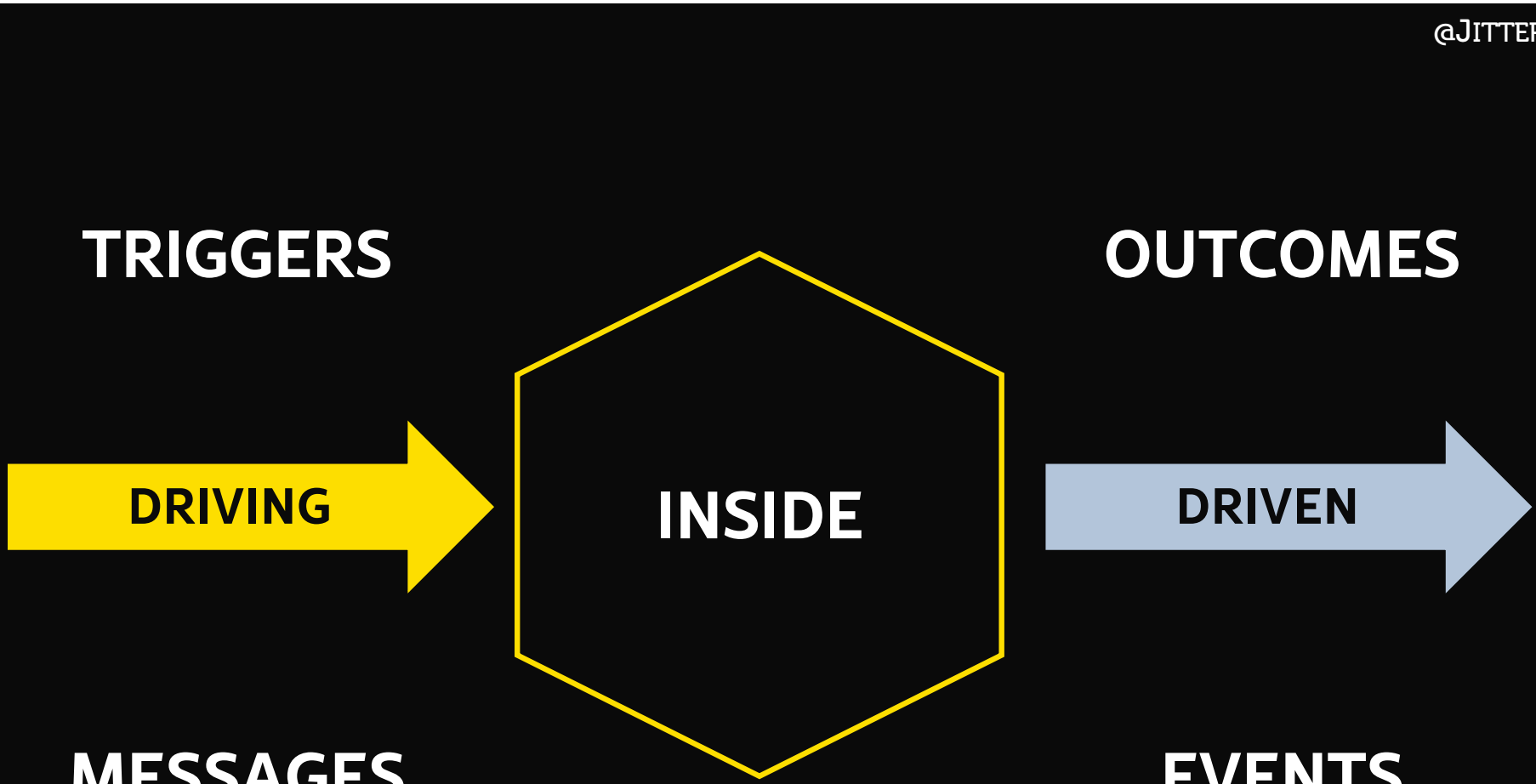
OUTSIDE

OUTSIDE

OUTSIDE

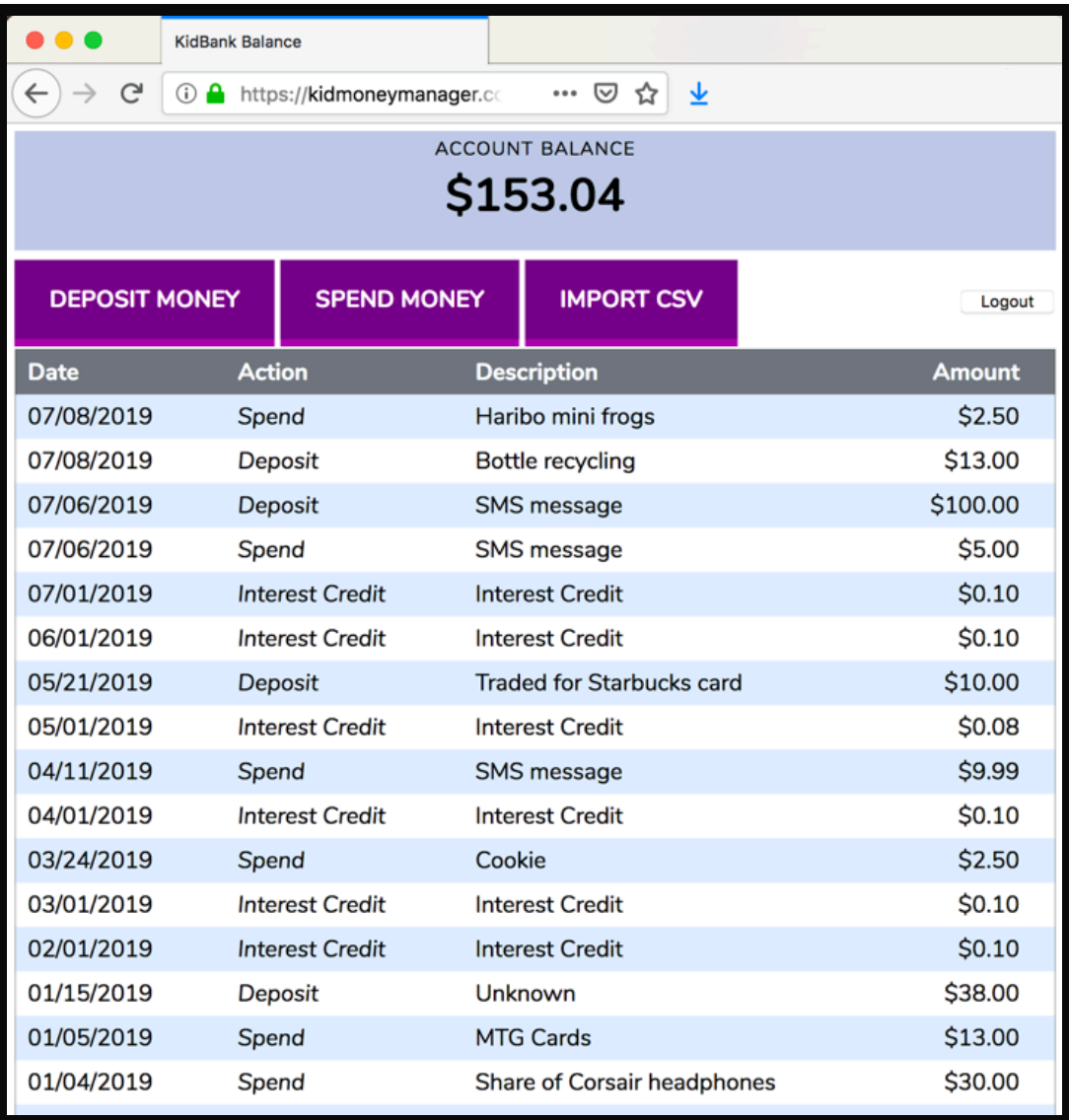
OUTSIDE

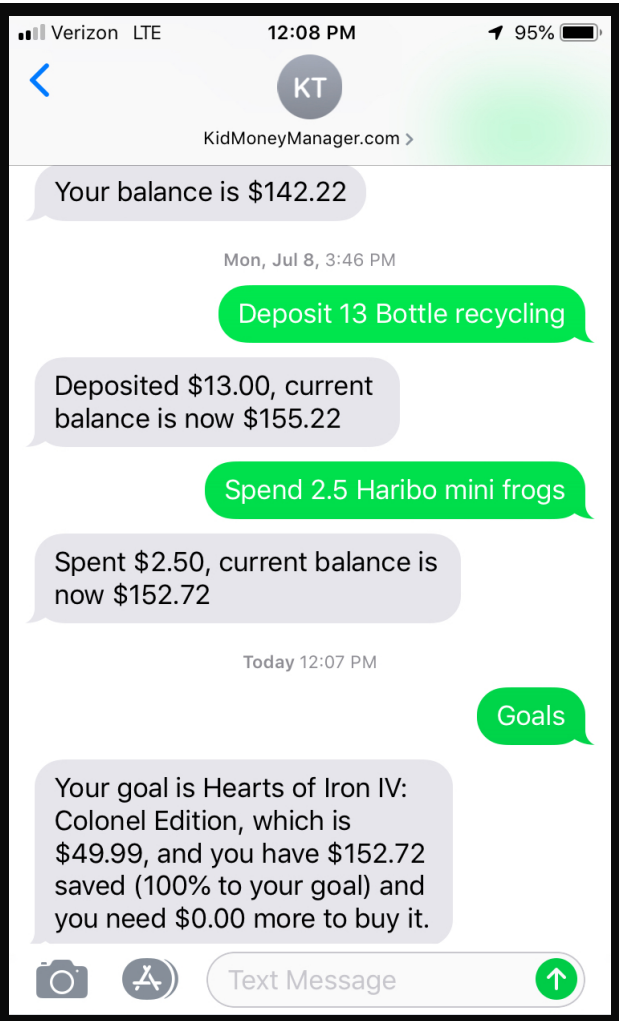
@JitterTed



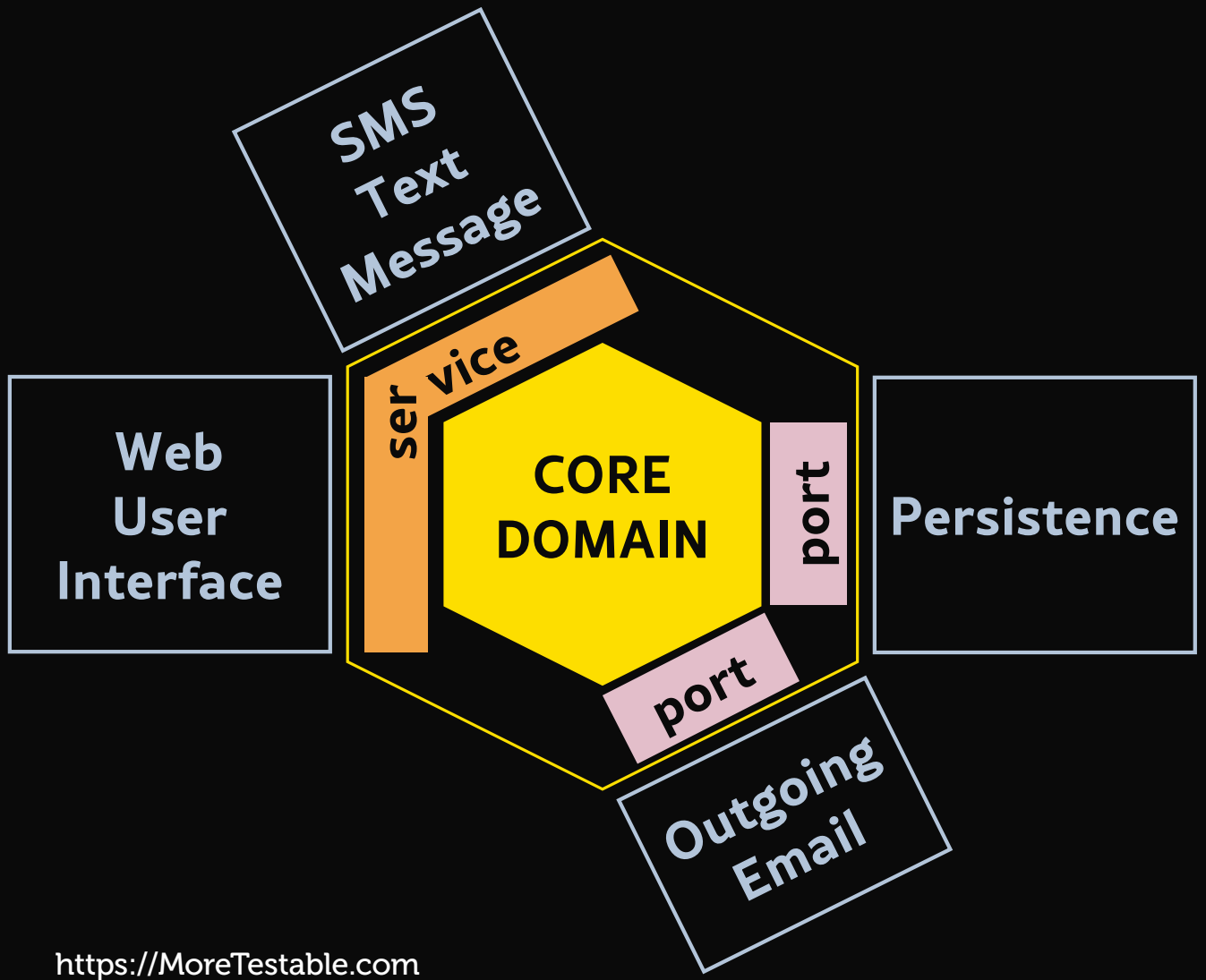
Ted M. Young https://MoreTestable.com

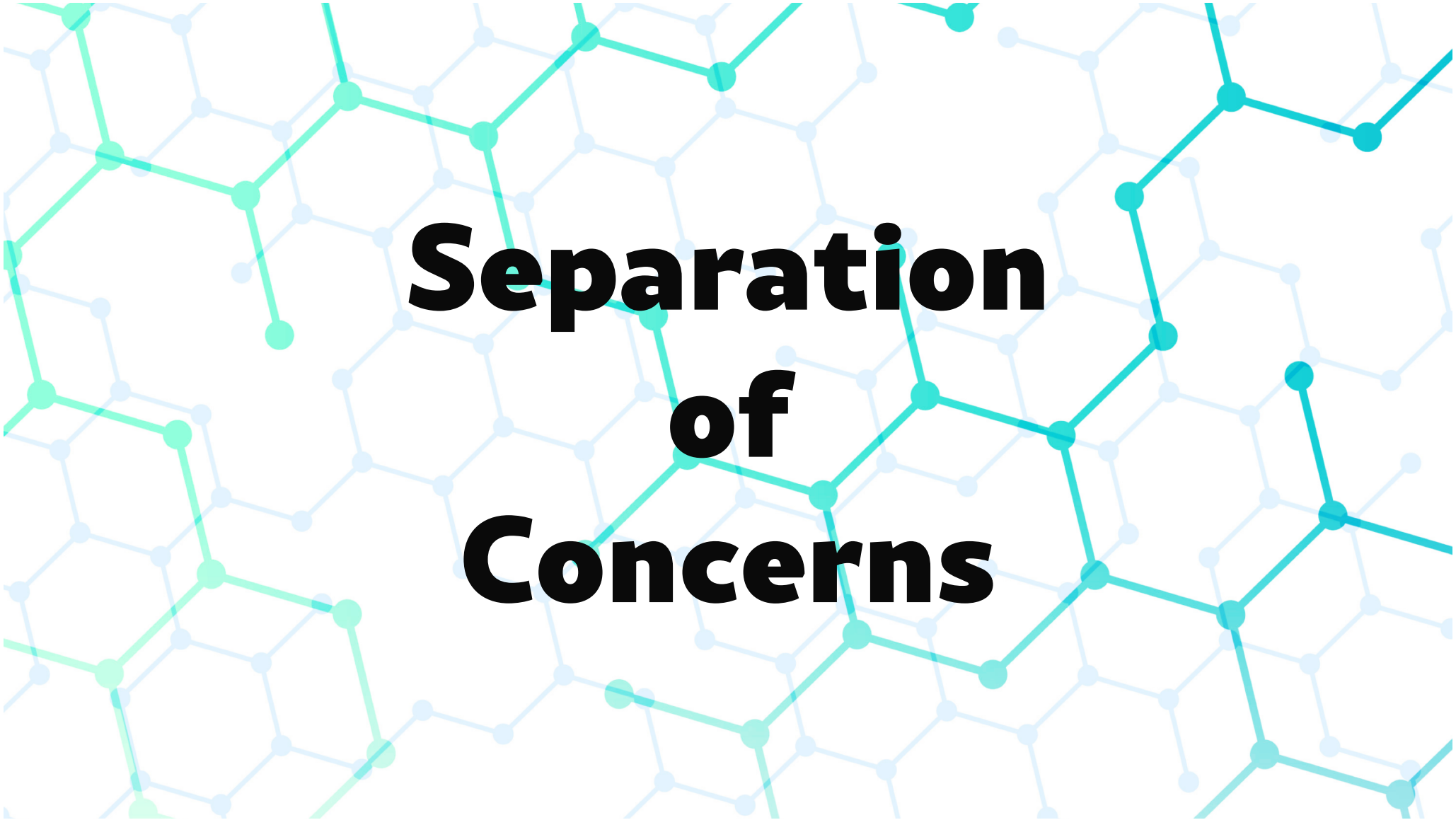
Live Coding: https://Twitch.tv/jitterted
Season 1 Videos: https://ted.dev/youtube
Source Code: https://github.com/tedyoung/kid-bank


KidBank Balance
https://kidmoneymanager.c
ACCOUNT BALANCE
$153.04
DEPOSIT MONEY
SPEND MONEY
IMPORT CSV
Logout
Date Action Description Amount
07/08/2019 Spend Haribo mini frogs $2.50
07/08/2019 Deposit Bottle recycling $13.00
07/06/2019 Deposit SMS message $100.00
07/06/2019 Spend SMS message $5.00
07/01/2019 Interest Credit Interest Credit $0.10
06/01/2019 Interest Credit Interest Credit $0.10
05/21/2019 Deposit Traded for Starbucks card $10.00
05/01/2019 Interest Credit Interest Credit $0.08
04/11/2019 Spend SMS message $9.99
04/01/2019 Interest Credit Interest Credit $0.10
03/24/2019 Spend Cookie $2.50
03/01/2019 Interest Credit Interest Credit $0.10
02/01/2019 Interest Credit Interest Credit $0.10
01/15/2019 Deposit Unknown $38.00
01/05/2019 Spend MTG Cards $13.00
01/04/2019 Spend Share of Corsair headphones $30.00


Verizon LTE
12:08 PM
95%
KT
KidMoneyManager.com
Your balance is $142.22
Mon, Jul 8, 3:46 PM
Deposit 13 Bottle recycling
Deposited $13.00, current balance is now $155.22
Spend 2.5 Haribo mini frogs
Spent $2.50, current balance is now $152.72
Today 12:07 PM
Goals
Your goal is Hearts of Iron IV: Colonel Edition, which is $49.99, and you have $152.72 saved (100% to your goal) and you need $0.00 more to buy it.
Text Message

@JITTERTED


SMS
Text
Message
Web
User
Interface
service
CORE
DOMAIN
port
Persistence
port
Outgoing
Email


Separation
of
Concerns

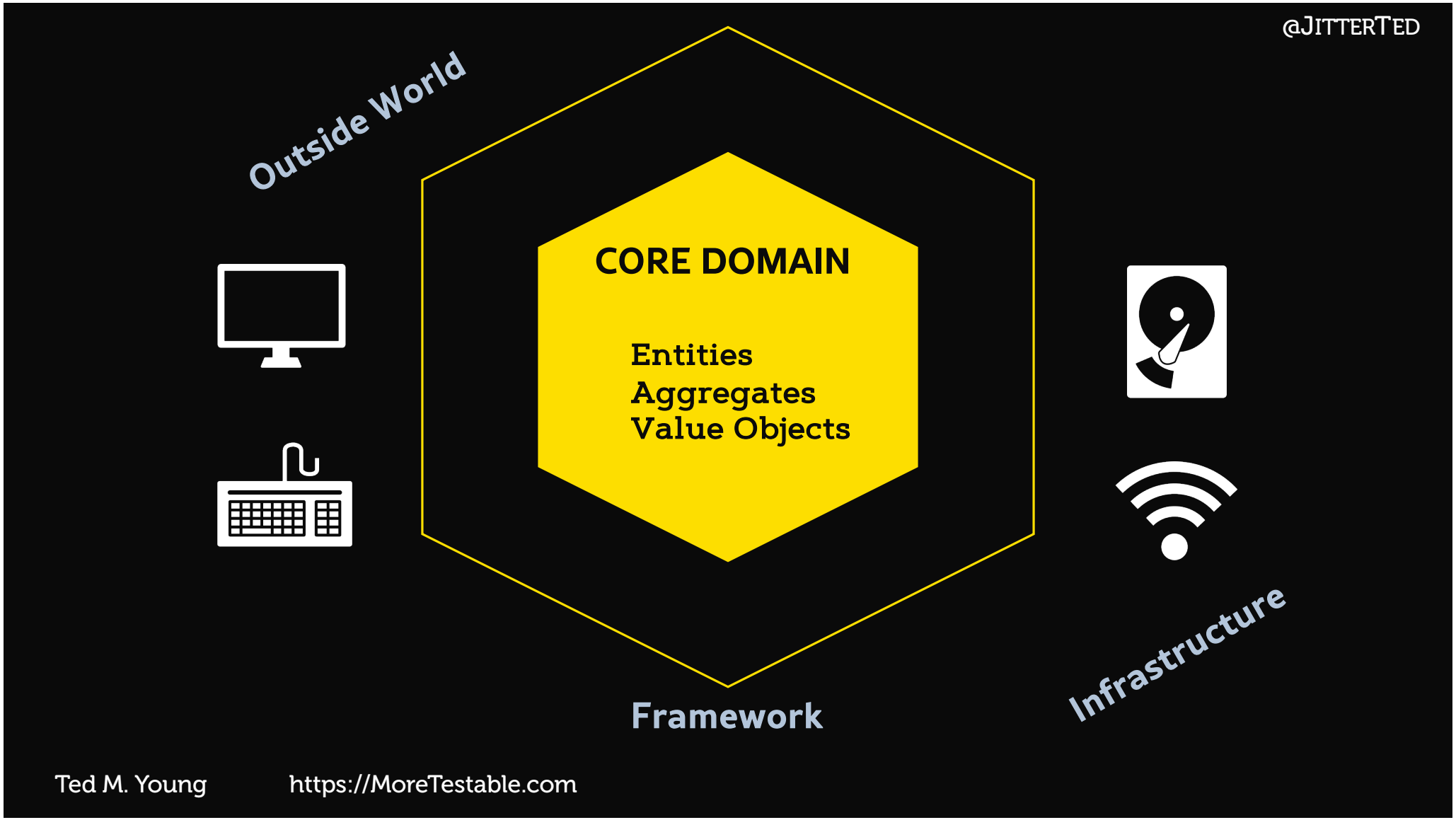
“[Separation of Concerns] is a matter about which a great deal is said and very little done.”

— Charles Dudley Warner

Principle

Separate Infrastructure and the Outside World from Business Logic

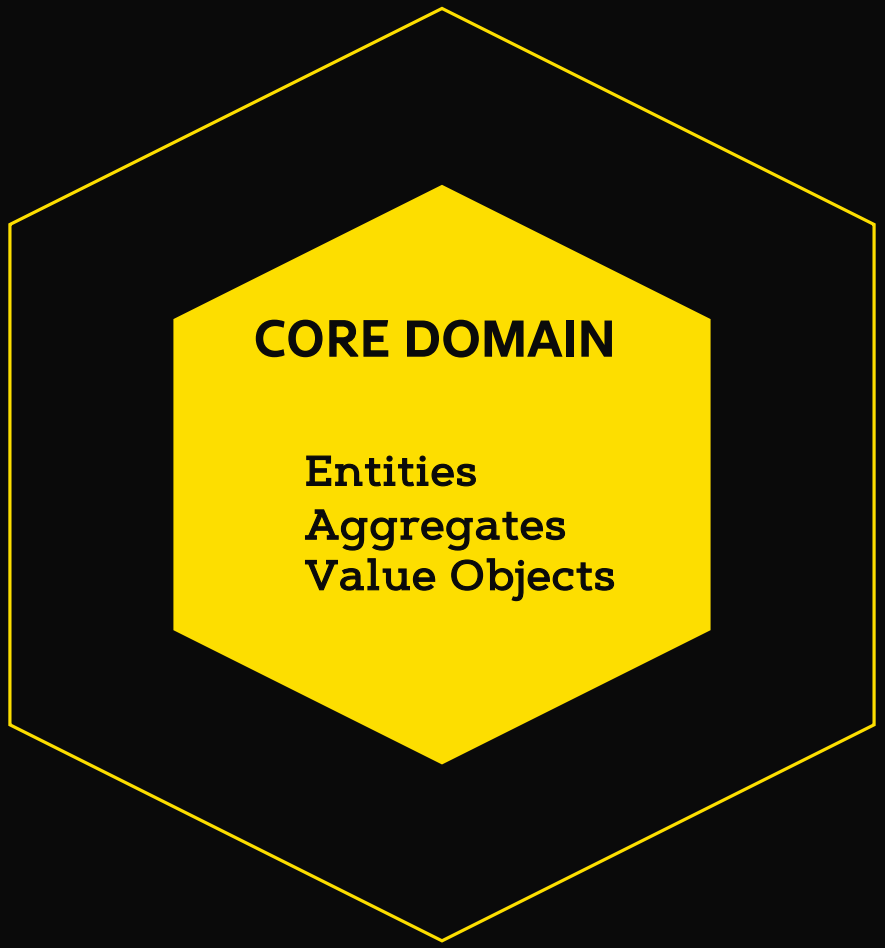
Ted M. Young https://MoreTestable.com


Outside World
CORE DOMAIN
Entities
Aggregates
Value Objects
Framework
Infrastructure

@JITTERTED

Listening to
Outside World

Console
HTTP API
Web UI
Messaging



Talking to
Infrastructure

Database
Network
Files

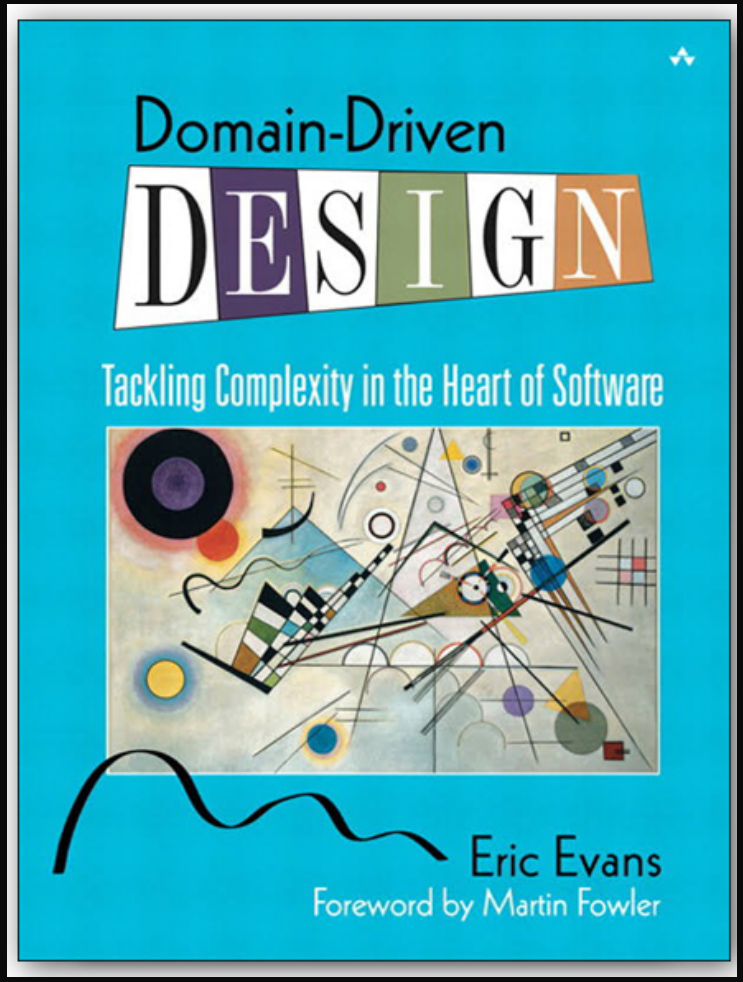
Frameworks **Spring, JEE, Micronaut**

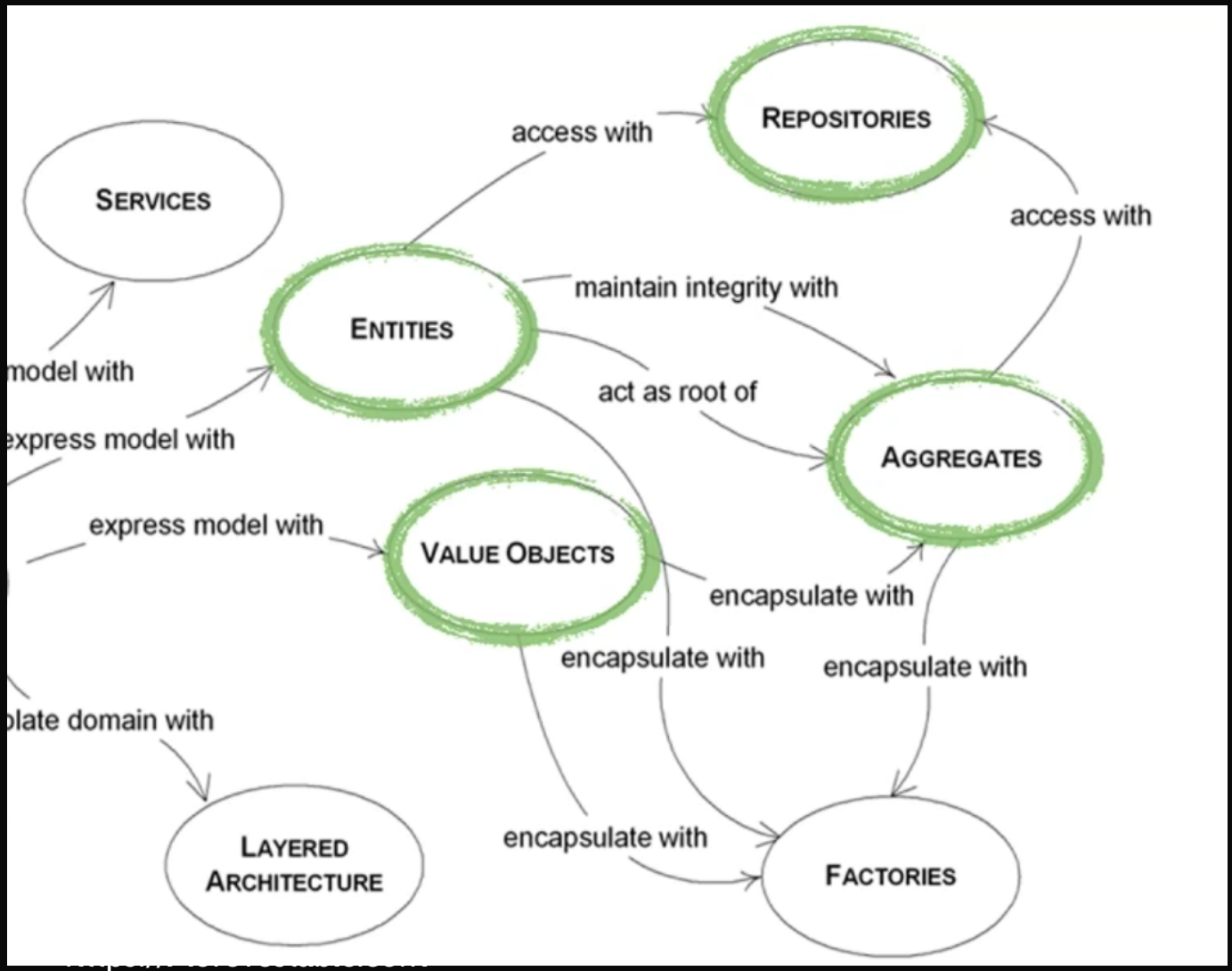
Ted M. Young https://MoreTestable.com

Domain-Driven

Domain is focus of the application

Ted M. Young https://MoreTestable.com


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler


REPOSITORIES
access with
SERVICES
access with
maintain integrity with
ENTITIES
model with
act as root of
AGGREGATES
express model with
express model with
VALUE OBJECTS
encapsulate with
encapsulate with
encapsulate with
late domain with
LAYERED ARCHITECTURE
encapsulate with
FACTORIES

Entity

Has Identity, History, and Attributes

Ted M. Young https://MoreTestable.com

Value Object

Has Only Attributes

Ted M. Young https://MoreTestable.com

@JitterTed

Aggregate

Entity that Enforces Consistency

Ted M. Young https://MoreTestable.com

@JitterTed

Repository

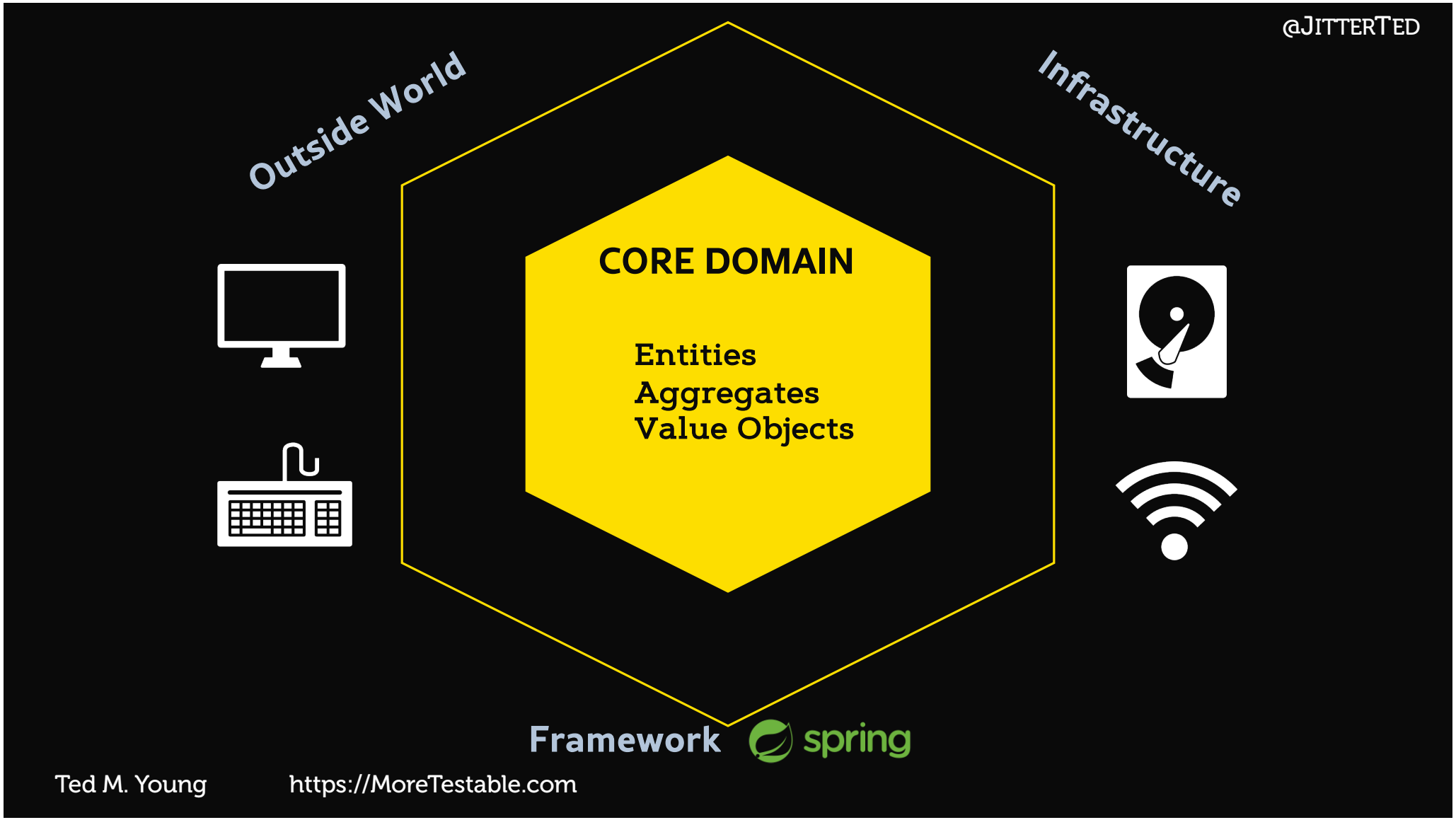
Use for Storing and Retrieving Aggregates

Ted M. Young https://MoreTestable.com

Core Domain

Account
SavingsGoal
PhoneNumber
Role

Transaction
UserProfile


@JitterTed
Outside World
Infrastructure
CORE DOMAIN
Entities
Aggregates
Value Objects
Framework
spring
Ted M. Young
https://MoreTestable.com

NOT a Unit Test

Talks to the database

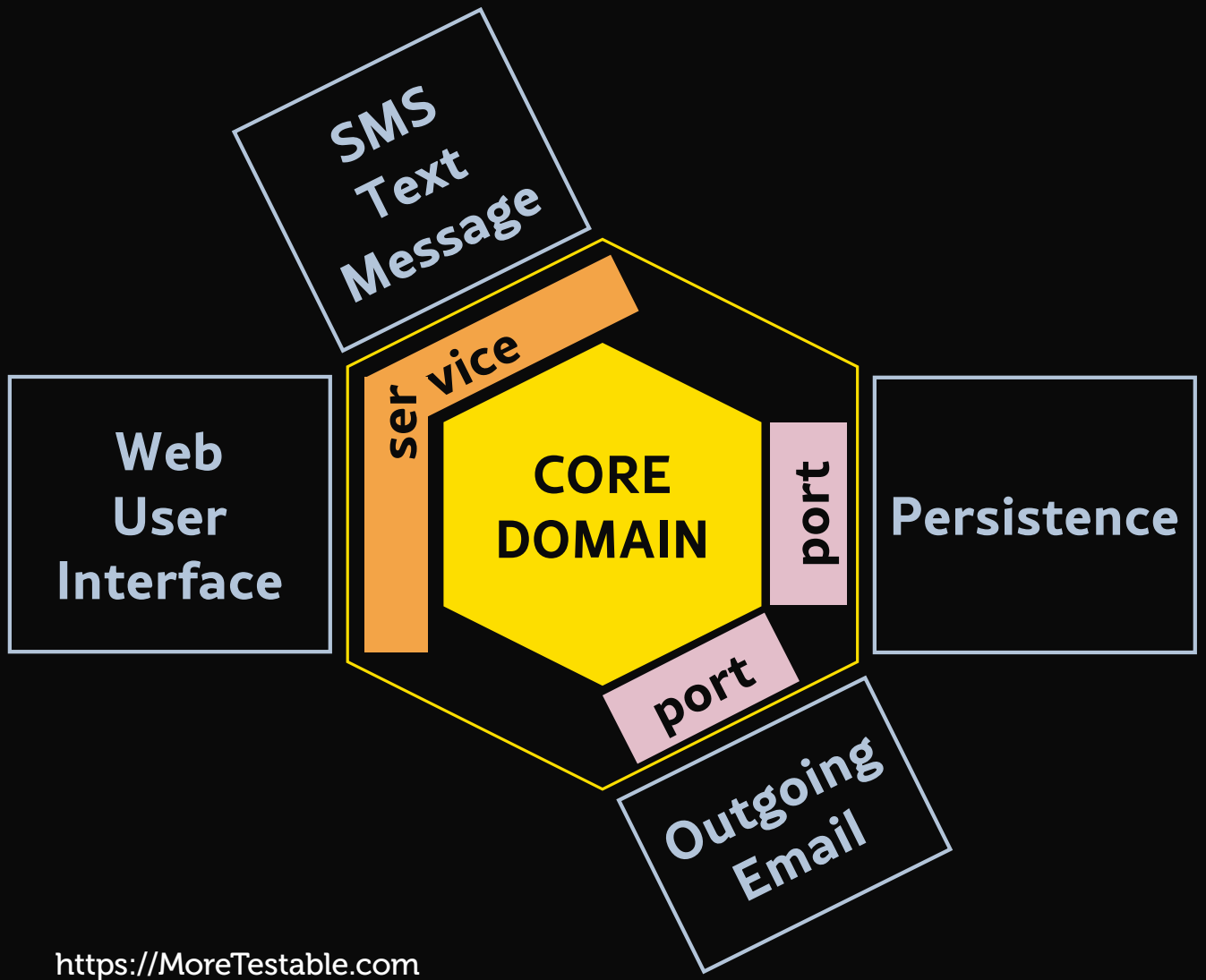
Communicates across a network

Touches the file system

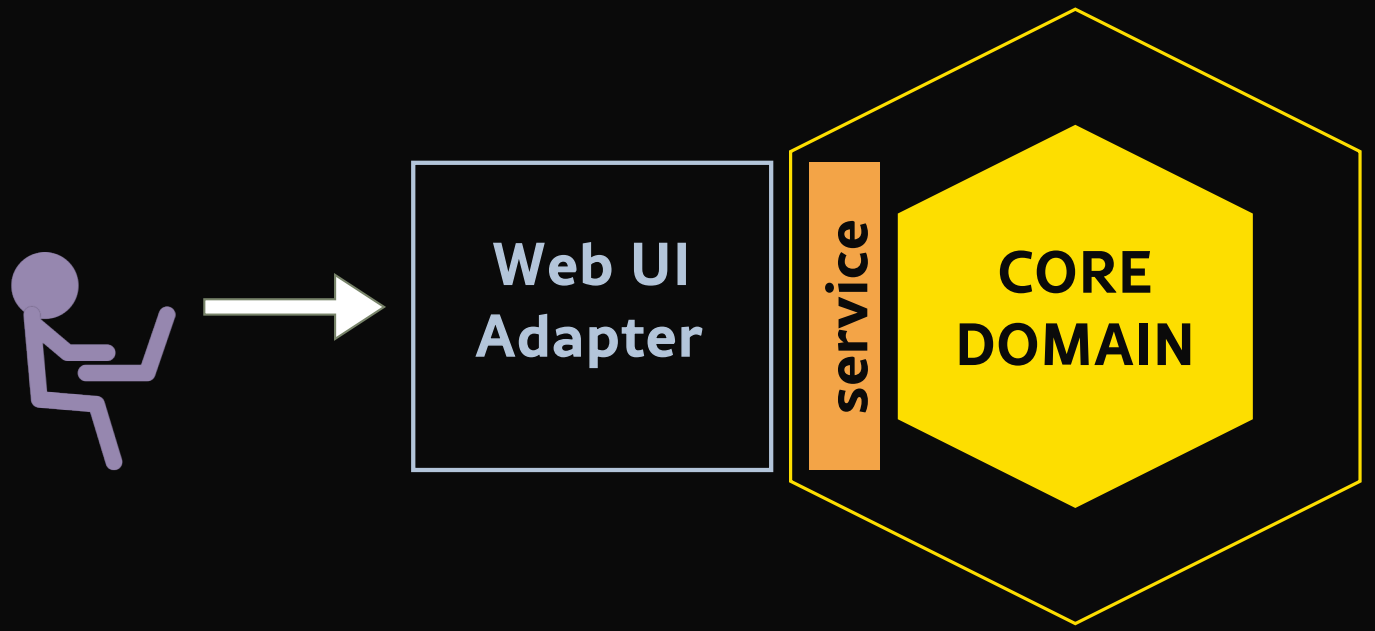
Michael Feathers
"A Set of Unit Testing Rules"
(2005)

Principle

Tests Must Be Isolated


SMS
Text
Message
Web
User
Interface
service
CORE
DOMAIN
port
Persistence
port
Outgoing
Email

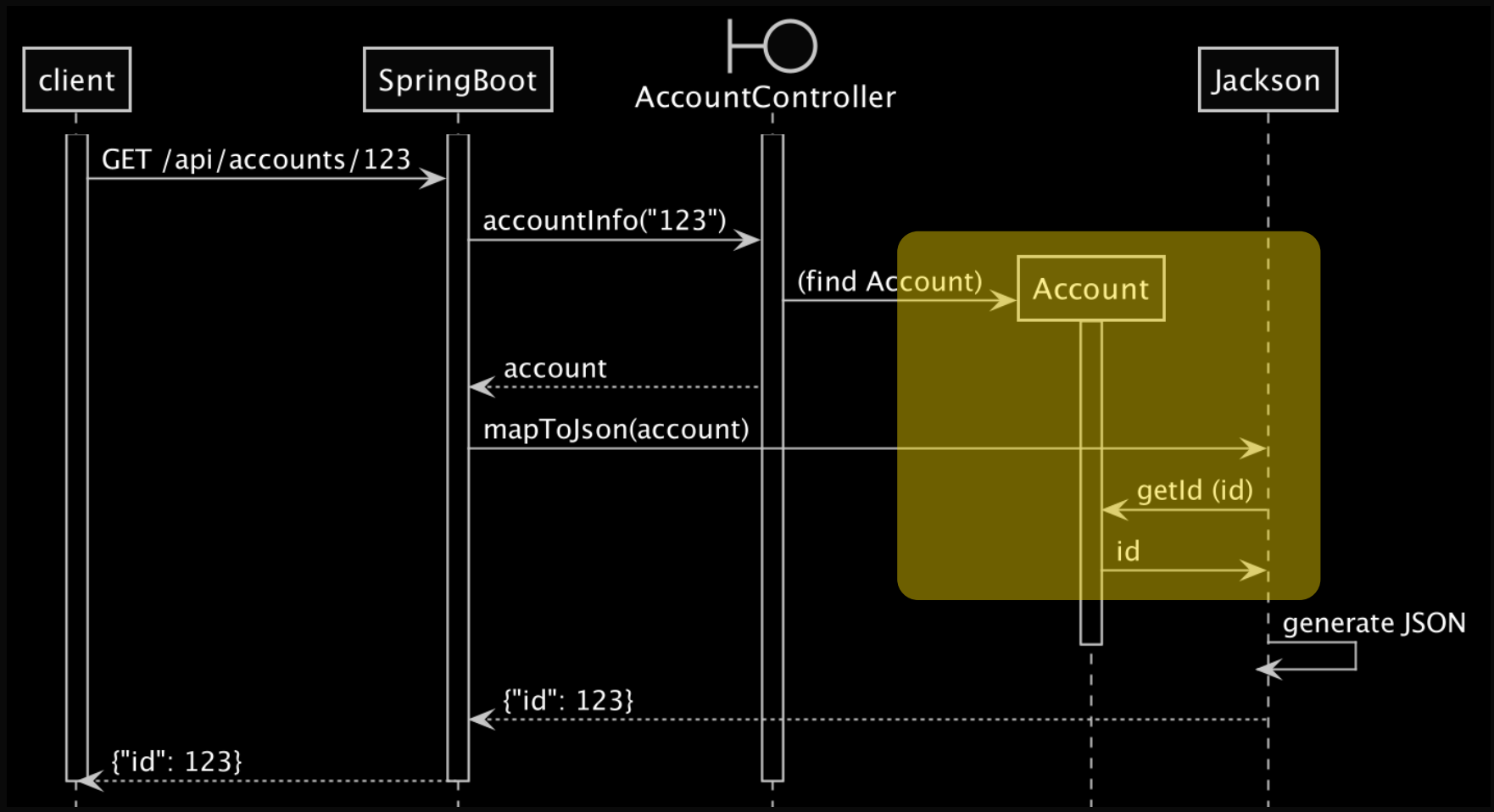
[Outside World]
User/Event-Driven



Domain Object (POJO)

```
public class Account {

  public void deposit(LocalDateTime txnDate, int amount, String descr){...}

  public void spend(LocalDateTime txnDate, int amount, String descr) {...}

  public int balance() {...}

}
```

@JITTERTED

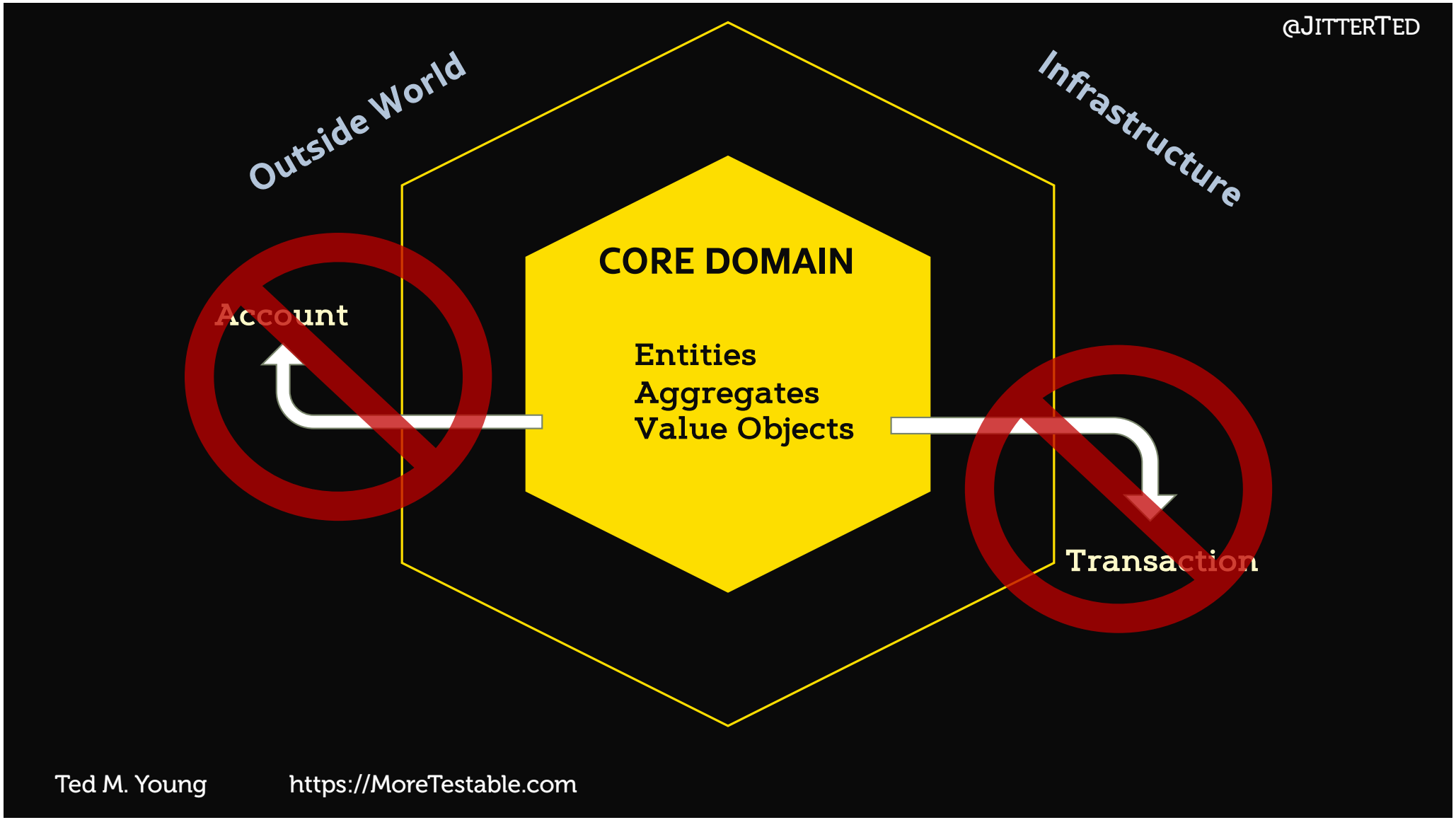


Ted M. Young https://MoreTestable.com

Rule

Domain Objects Never Leave

Ted M. Young https://MoreTestable.com


@JitterTed
Outside World
Infrastructure
CORE DOMAIN
Entities
Aggregates
Value Objects
Account
Transaction
Ted M. Young
https://MoreTestable.com

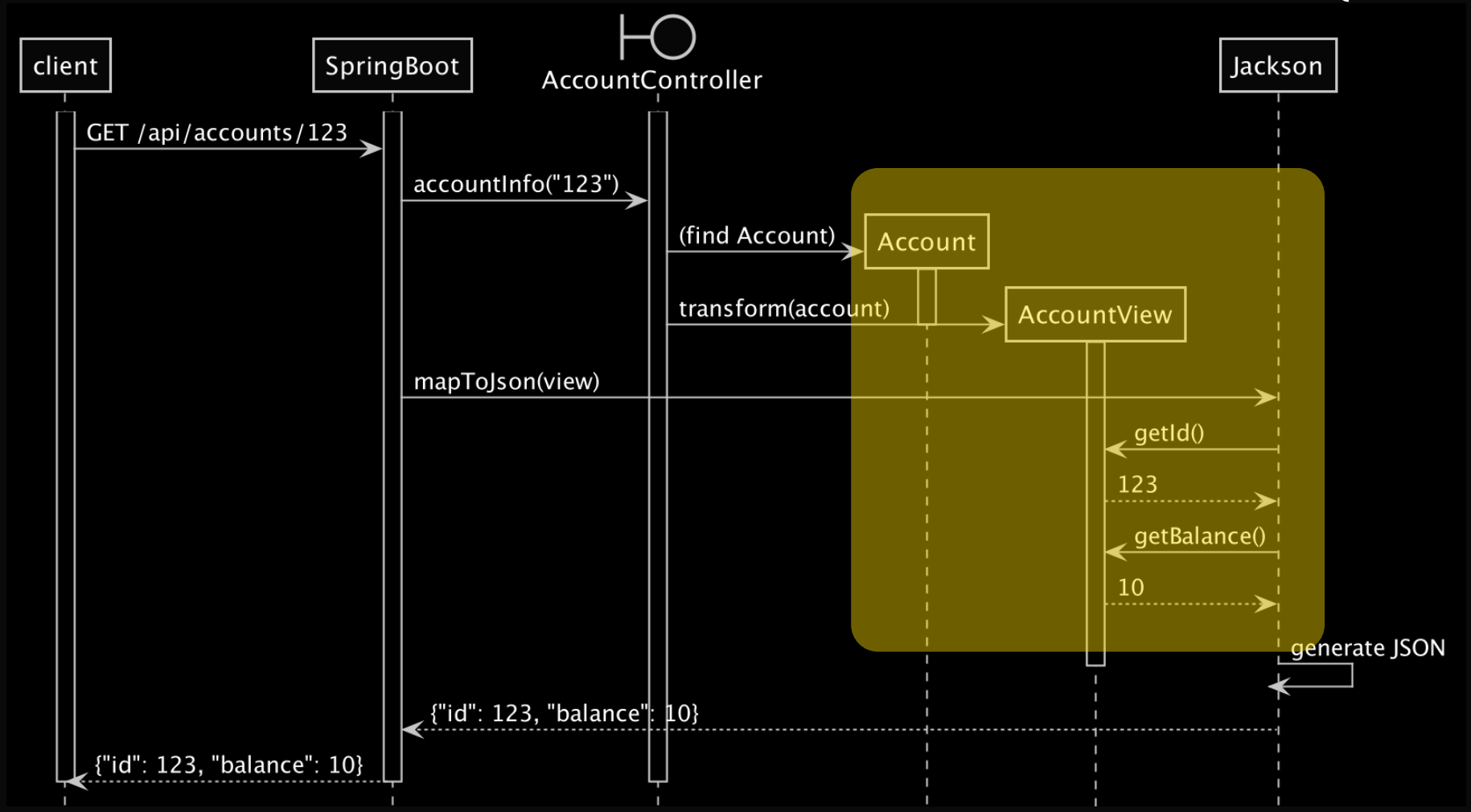
Goal

Reduce Coupling

Data Transfer Object (DTO)

```
public class AccountView {

  private long id;

  private int balance;


 /*
  * setters & getters here
  */
}
```

@JitterTed



Ted M. Young https://MoreTestable.com

Adapter Role

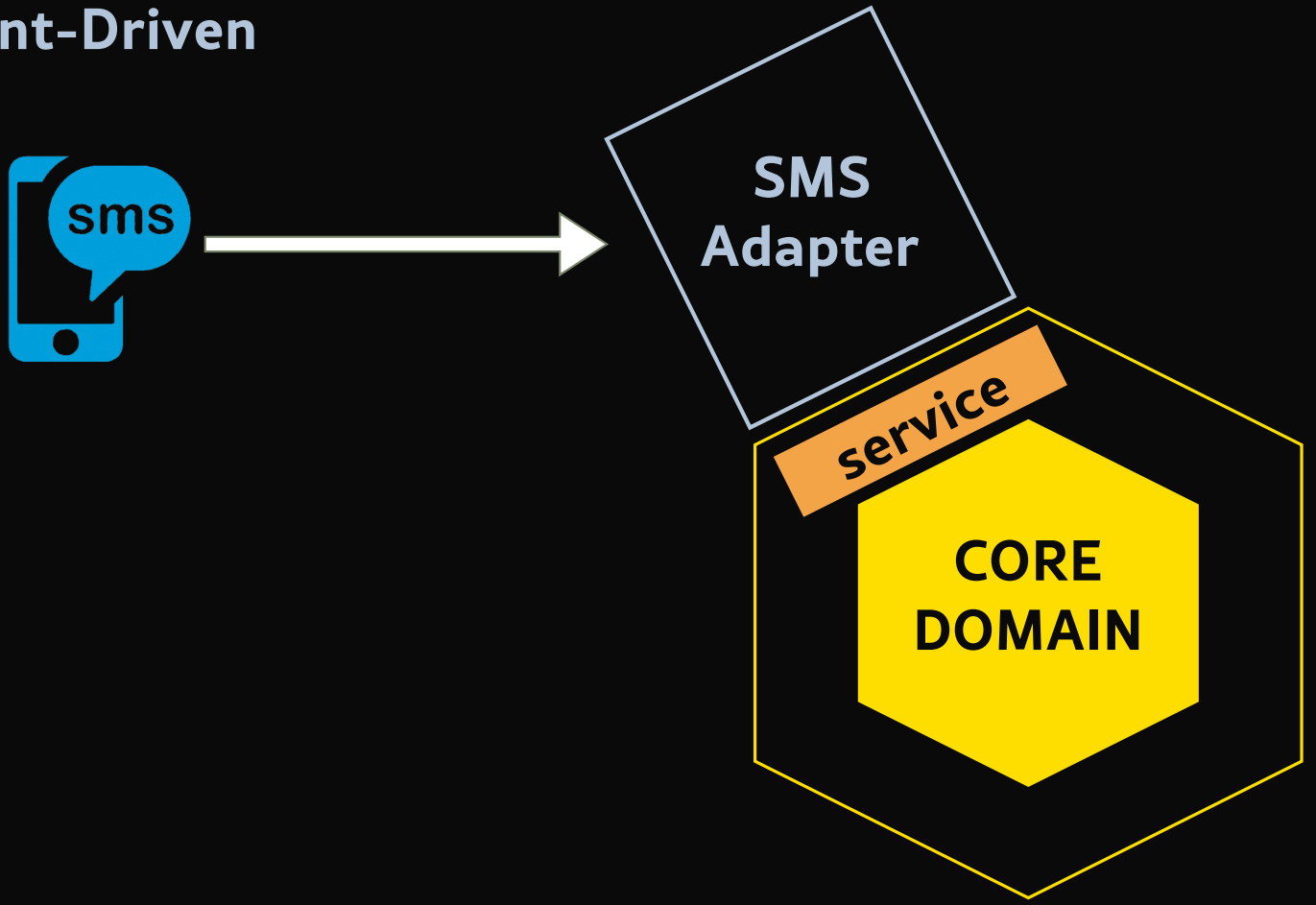
Package of Classes

Ted M. Young https://MoreTestable.com

Adapter Responsibilities

1. Validate input from outside-world
2. Convert to Domain Object
3. Send to Service to do work *(optional)*
4. Domain Object —> DTO
5. Return DTO to Outside-World

[Outside World]
User/Event-Driven

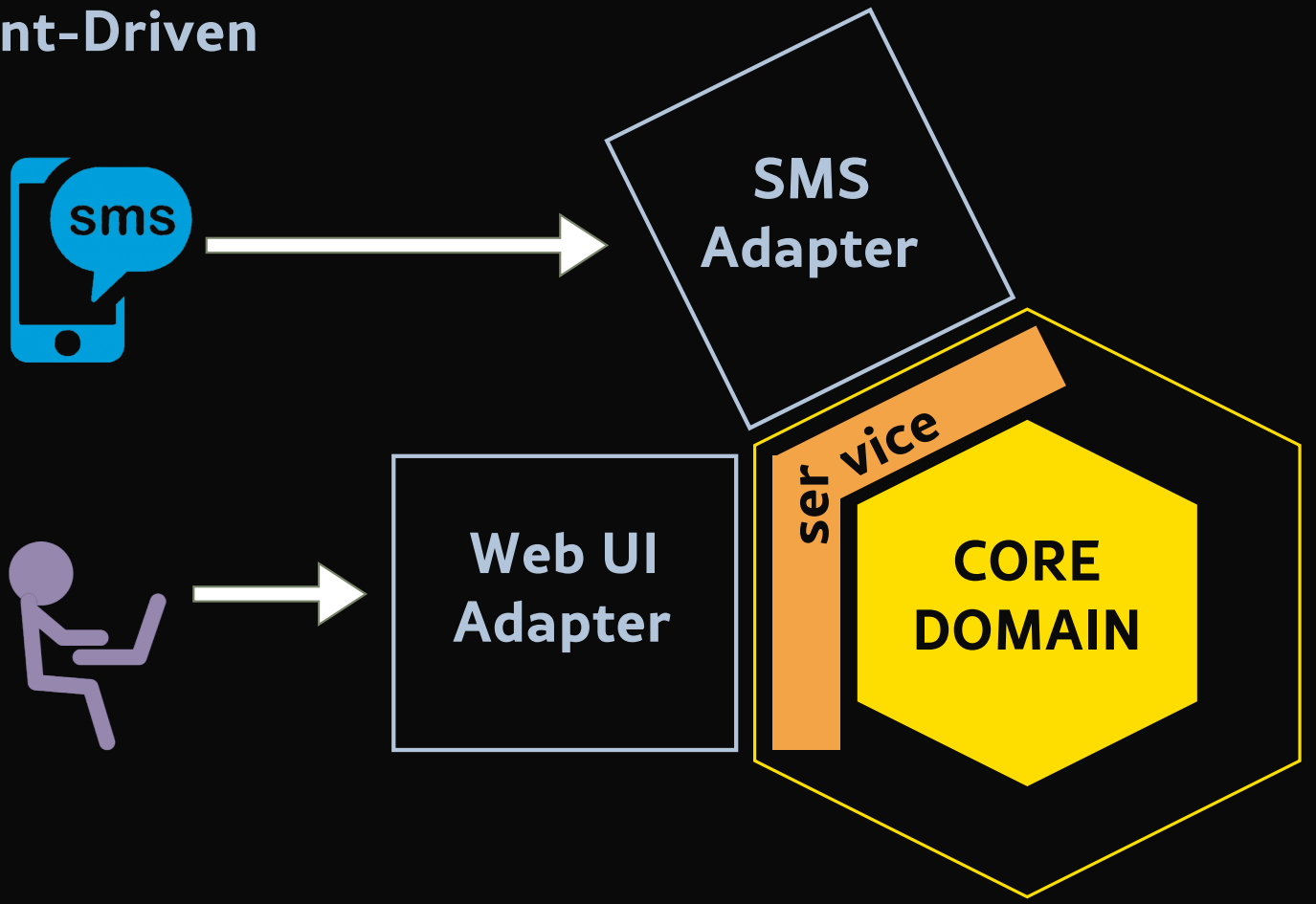


Rule

Adapter Per Communication Mechanism

Ted M. Young https://MoreTestable.com

[Outside World]
User/Event-Driven



@JitterTed

Dependency Rule

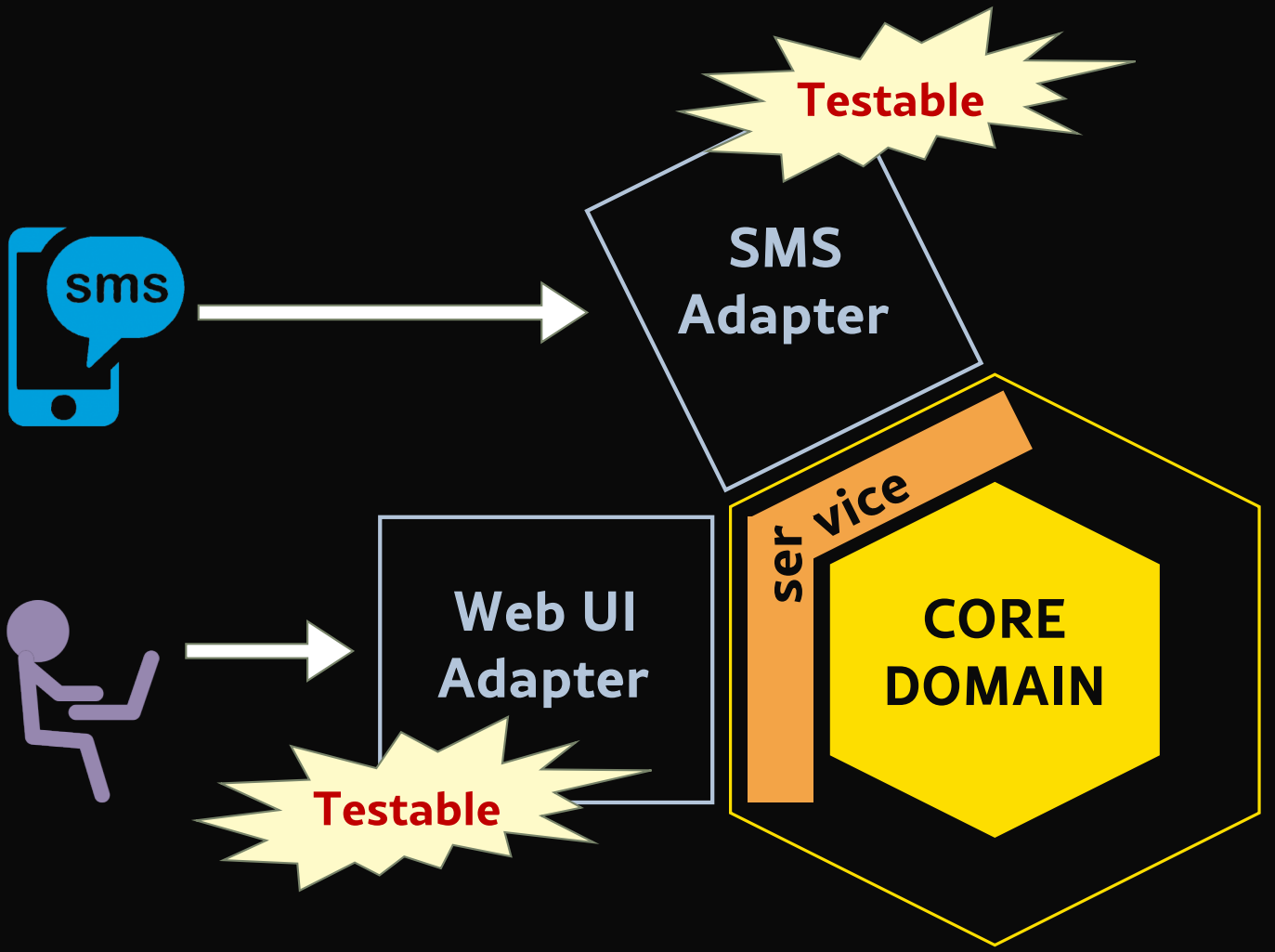
Adapters Don't Depend on Adapters

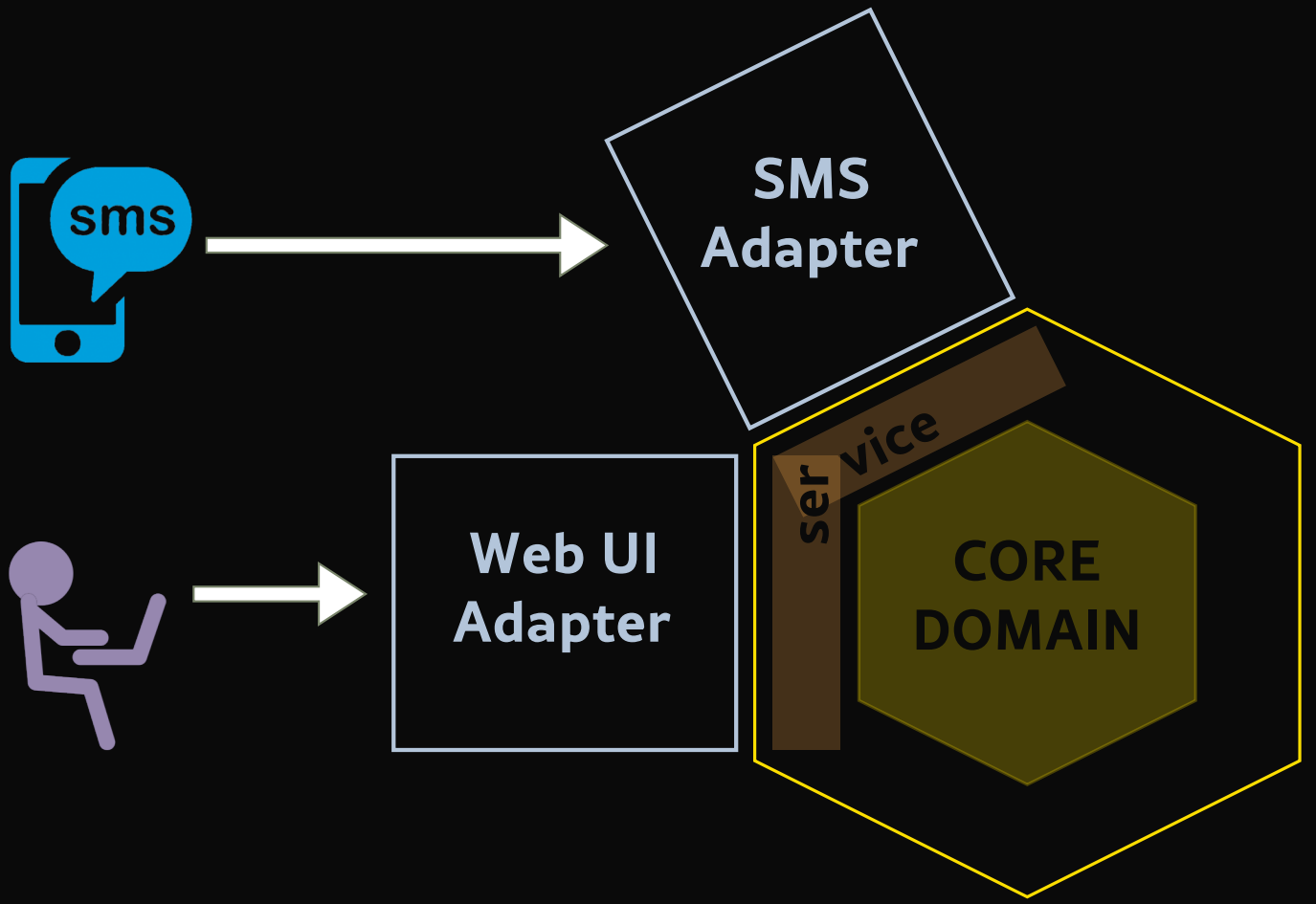
Ted M. Young https://MoreTestable.com

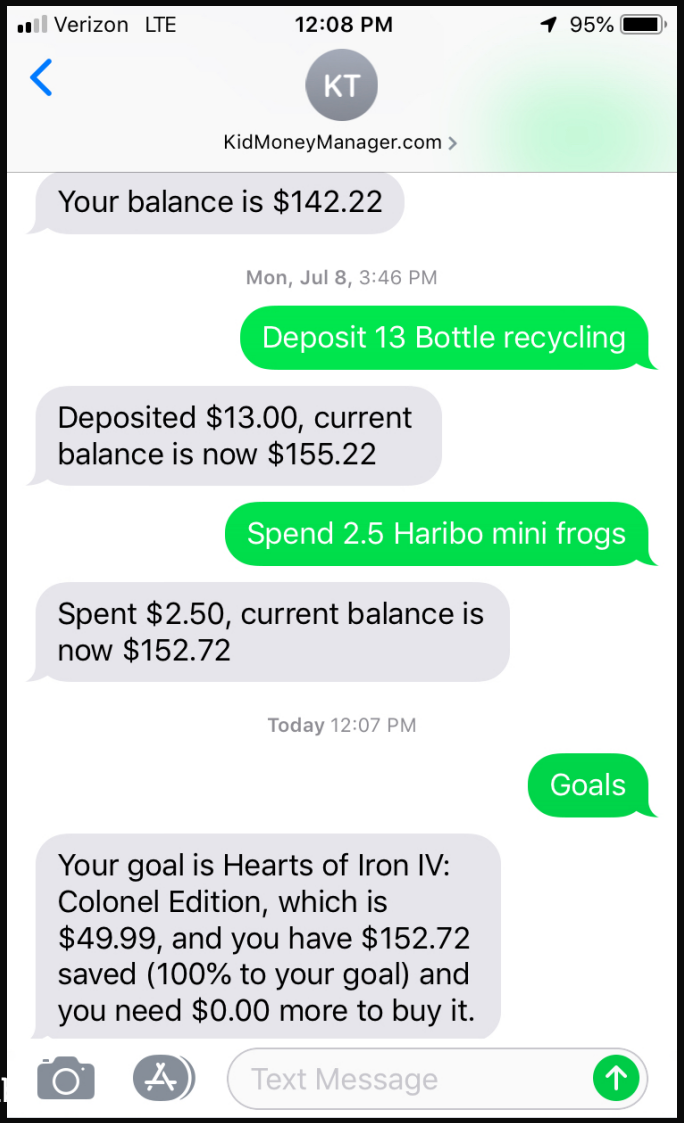
@JitterTed

Testing Adapters

Ted M. Young https://MoreTestable.com

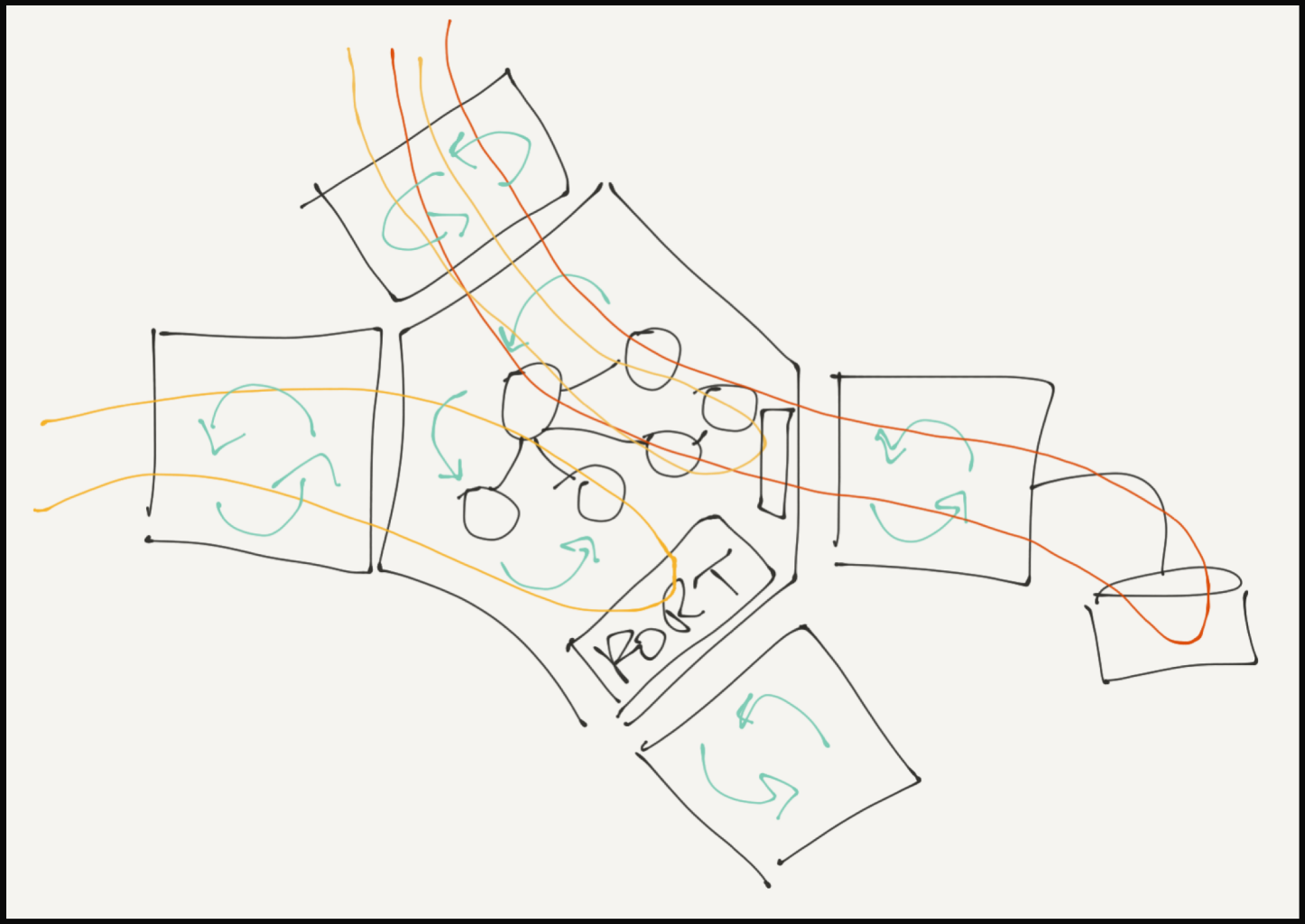

Testable
SMS
Adapter
sms
ser vice
CORE
DOMAIN
Web UI
Adapter
Testable


sms
SMS Adapter
Web UI Adapter
service
CORE DOMAIN


Verizon LTE
12:08 PM
95%
KT
KidMoneyManager.com
Your balance is $142.22
Mon, Jul 8, 3:46 PM
Deposit 13 Bottle recycling
Deposited $13.00, current balance is now $155.22
Spend 2.5 Haribo mini frogs
Spent $2.50, current balance is now $152.72
Today 12:07 PM
Goals
Your goal is Hearts of Iron IV: Colonel Edition, which is $49.99, and you have $152.72 saved (100% to your goal) and you need $0.00 more to buy it.
Text Message

```java
public class CommandParserSpendCommandTest {

  @Test
  public void spendWithSingleWordDescriptionCreatesSpendCommandWithDescription() throws Exception {
    CoreAccount account = null;
    CommandParser commandParser = new CommandParser(account);

    TransactionCommand command = commandParser.parse("spend 37.95 Cards");

    assertThat(command)
        .isEqualTo(new SpendCommand(account, 37_95, "Cards"));
  }

  @Test
  public void spendOnlyWithValidAmountShouldBeSpendCommand() throws Exception {
    CoreAccount account = null;
    CommandParser commandParser = new CommandParser(account);

    TransactionCommand command = commandParser.parse("spend 55");

    SpendCommand expectedSpendCommand = new SpendCommand(account, 55_00);

    assertThat(command)
        .isEqualTo(expectedSpendCommand);
  }
```

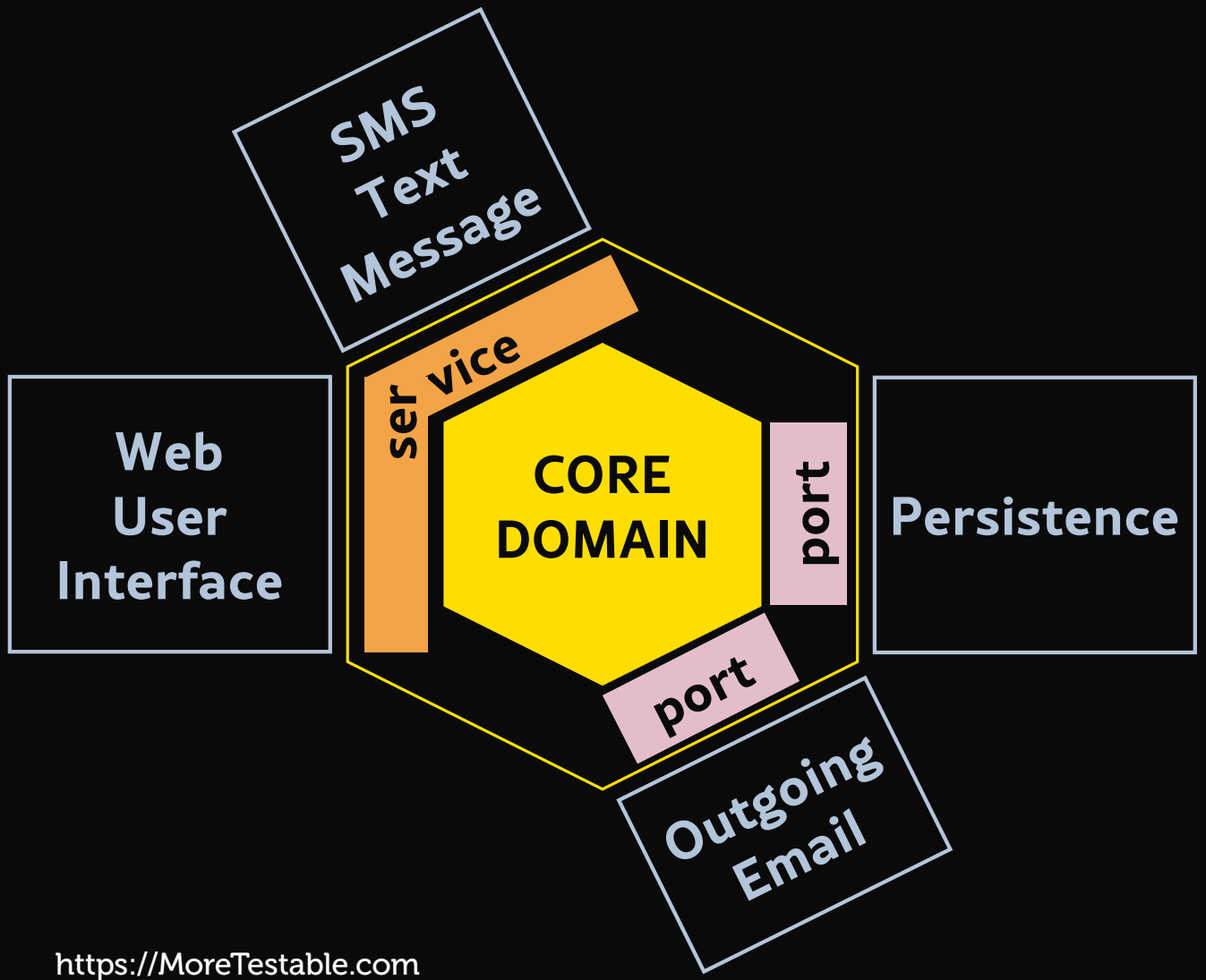
```java
public class DepositViewIntegrationTest {
  @Autowired
  private MockMvc mockMvc;

  @Test
  public void getAgainstDepositUriShouldReturnDepositFormPage() throws Exception {
    mockMvc.perform(get("/deposit"))
           .andExpect(status().isOk())
           .andExpect(view().name("deposit"))
           .andExpect(model().attributeExists("balance", "depositCommand"));
  }

  @Test
  public void submitDepositFormRedirectsToHomePage() throws Exception {
    mockMvc.perform(post("/deposit")
                        .param("amount", "12.45")
                        .param("date", "2000-01-02")
                        .param("description", "the source of money"))
           .andExpect(redirectedUrl(AccountController.ACCOUNT_URL));

    mockMvc.perform(get(AccountController.ACCOUNT_URL))
           .andExpect(status().isOk());
  }
```


PORT

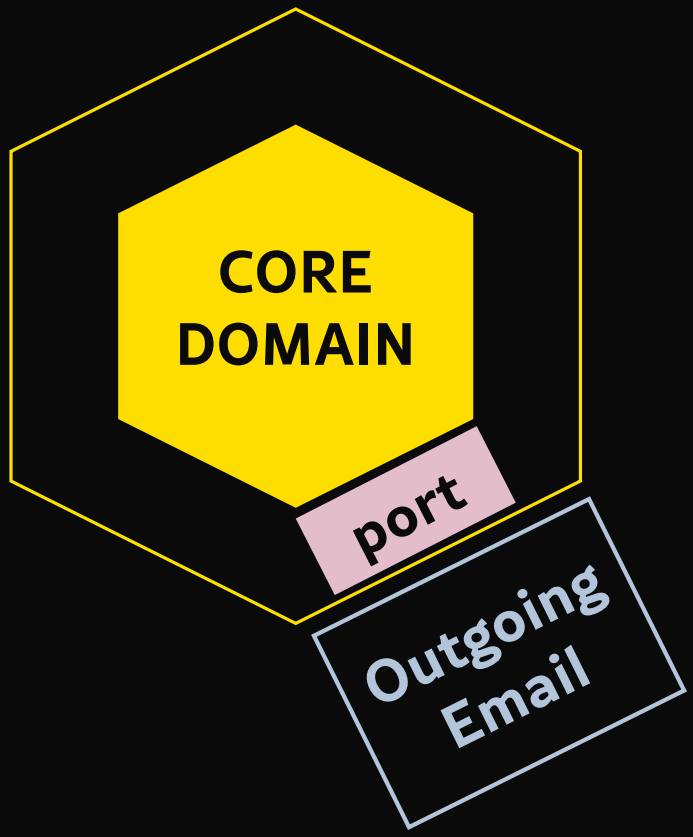
Ports

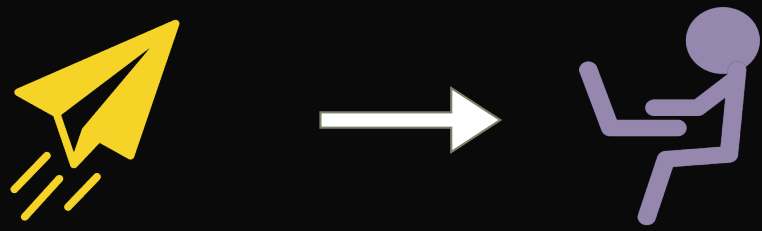
Abstractions and Concrete Implementations

Ted M. Young https://MoreTestable.com


SMS
Text
Message
Web
User
Interface
service
CORE
DOMAIN
port
Persistence
port
Outgoing
Email

[Outside World]
Application-Driven





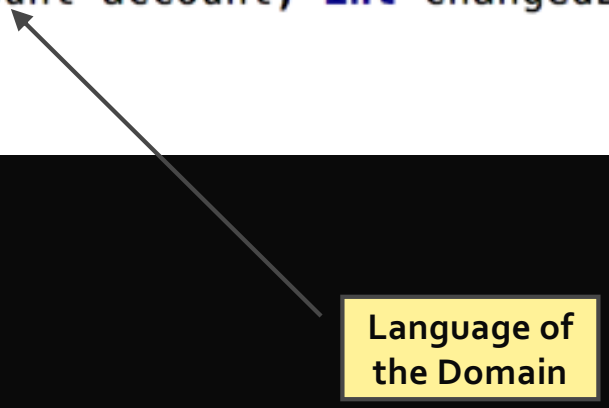
Step 1: Define Port

Ideal Interface

in language of Domain

Ted M. Young https://MoreTestable.com

```
package com.learnwithted.kidbank.domain;

public interface BalanceChangedNotifier {
  void balanceChanged(Account account, int changedBy);
}
```



Step 2

Use Spy to Test

```java
public class BalanceChangeNotificationTest {
  @Test
  public void depositShouldCauseNotificationOfBalanceChange() throws Exception {
    BalanceChangedNotifier mockNotifier = mock(BalanceChangedNotifier.class);

    // Given an account
    Account account = TestAccountBuilder.builder()
                                        .notifier(mockNotifier)
                                        .buildAsCore();

    // When we deposit money
    account.deposit(now(), 25_00, "test");

    // Then we expect the notification to be sent
    verify(mockNotifier).balanceChanged(account, 25_00);
  }
```

technically a spy

validation

Step 3: Create Implementation

Concrete Adapter

```java
package com.learnwithted.kidbank.adapter.sms;

@Service
public class TextMessageBalanceChangedNotifier implements BalanceChangedNotifier {

  private final TextMessageSender textMessageSender;

  public TextMessageBalanceChangedNotifier(TextMessageSender textMessageSender) {
    this.textMessageSender = textMessageSender;
  }

  @Override
  public void balanceChanged(Account account, int changedBy) {
    String to = "+16505551212";
    String body = "Balance changed by "
                    + ScaledDecimals.formatAsMoney(changedBy)
                    + ", current balance is "
                    + ScaledDecimals.formatAsMoney(account.balance());
    textMessageSender.send(to, body);
  }
}
```

SMS adapter

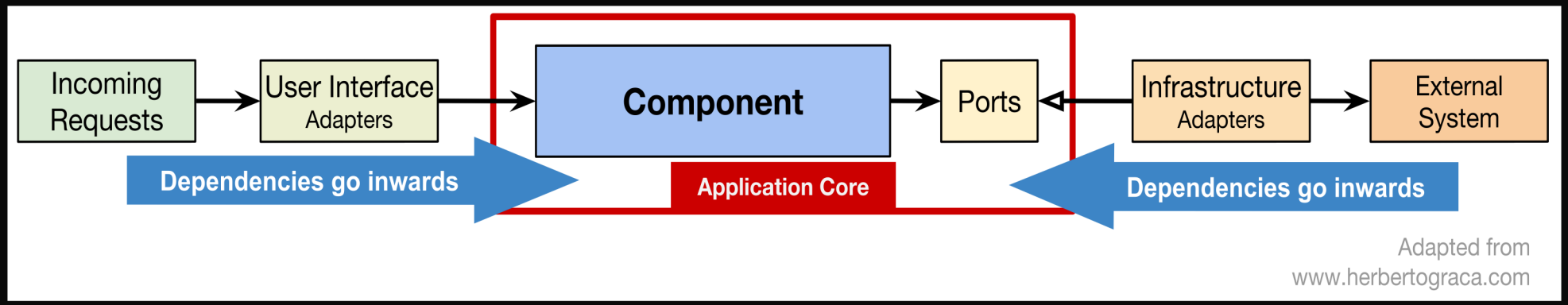
Null Object Pattern

```
package com.learnwithted.kidbank.domain;

public class DoNothingBalanceChangeNotifier implements BalanceChangedNotifier {
  @Override
  public void balanceChanged(Account account, int changedBy) {
    // do nothing
  }
}
```

Dependency Rule

Dependencies Point Inwards

Ted M. Young https://MoreTestable.com


Incoming Requests
User Interface Adapters
Component
Ports
Infrastructure Adapters
External System
Dependencies go inwards
Application Core
Dependencies go inwards
Adapted from
www.herbertograca.com

Rule

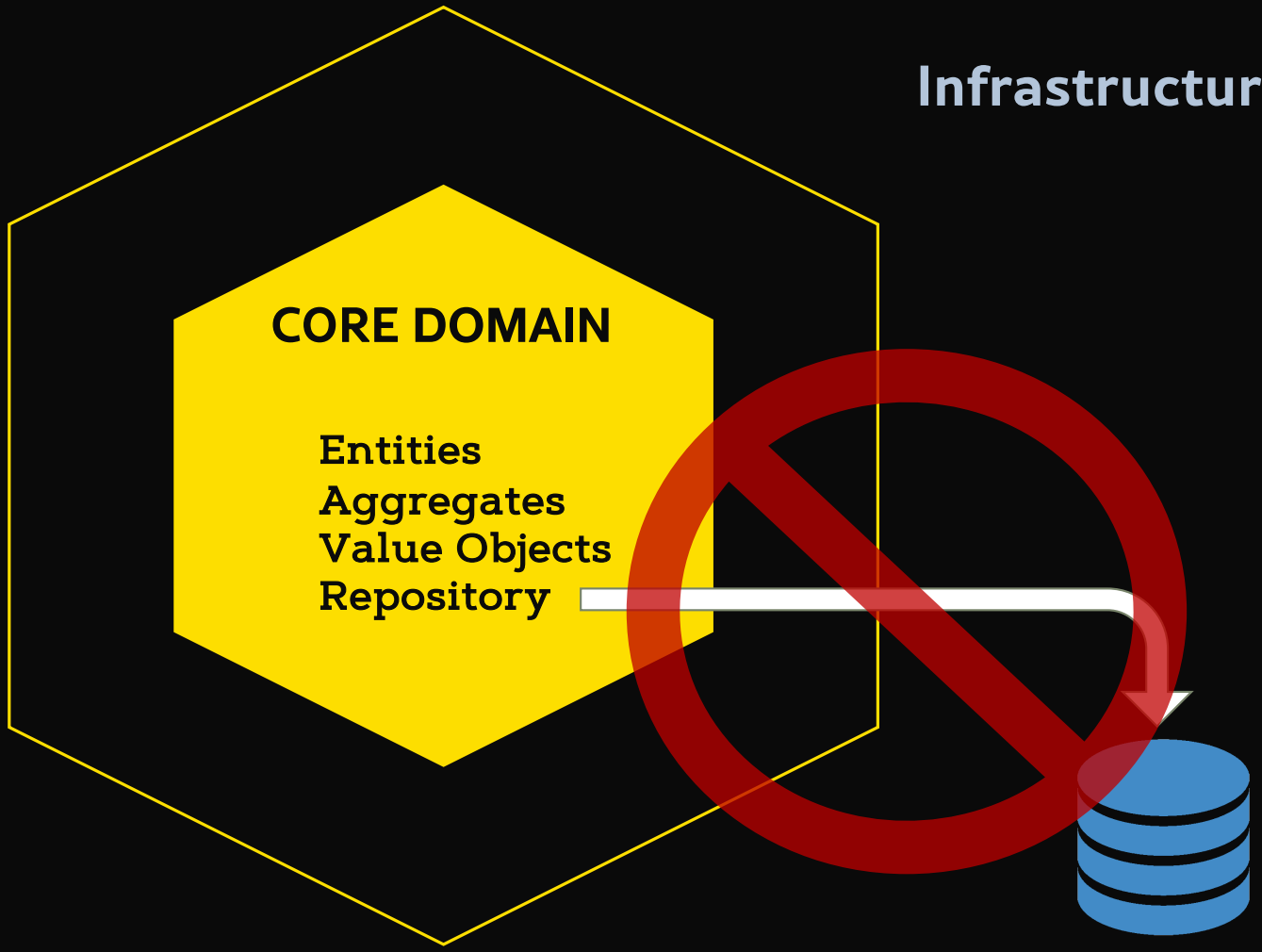
Persistence Ignorance

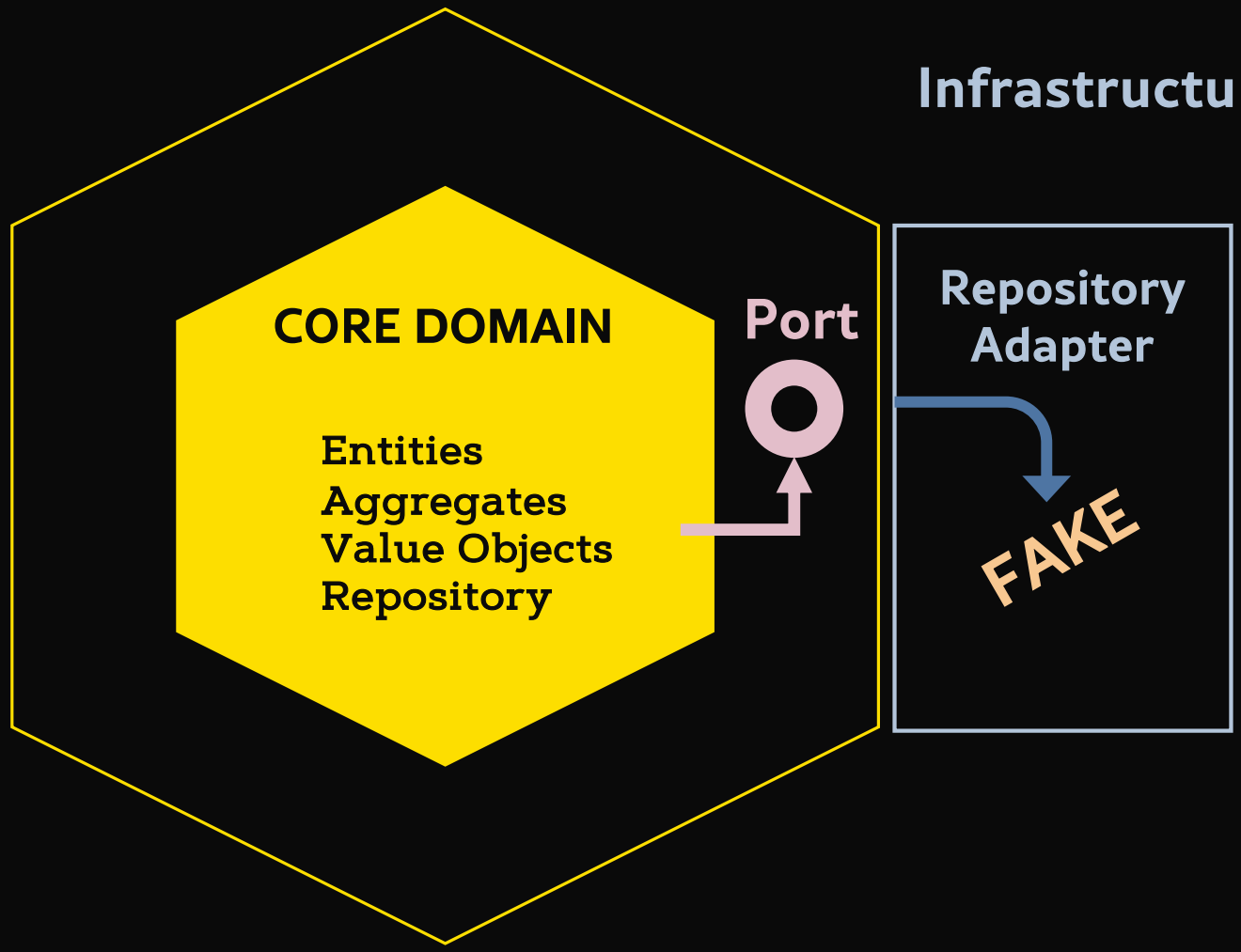
Core Domain

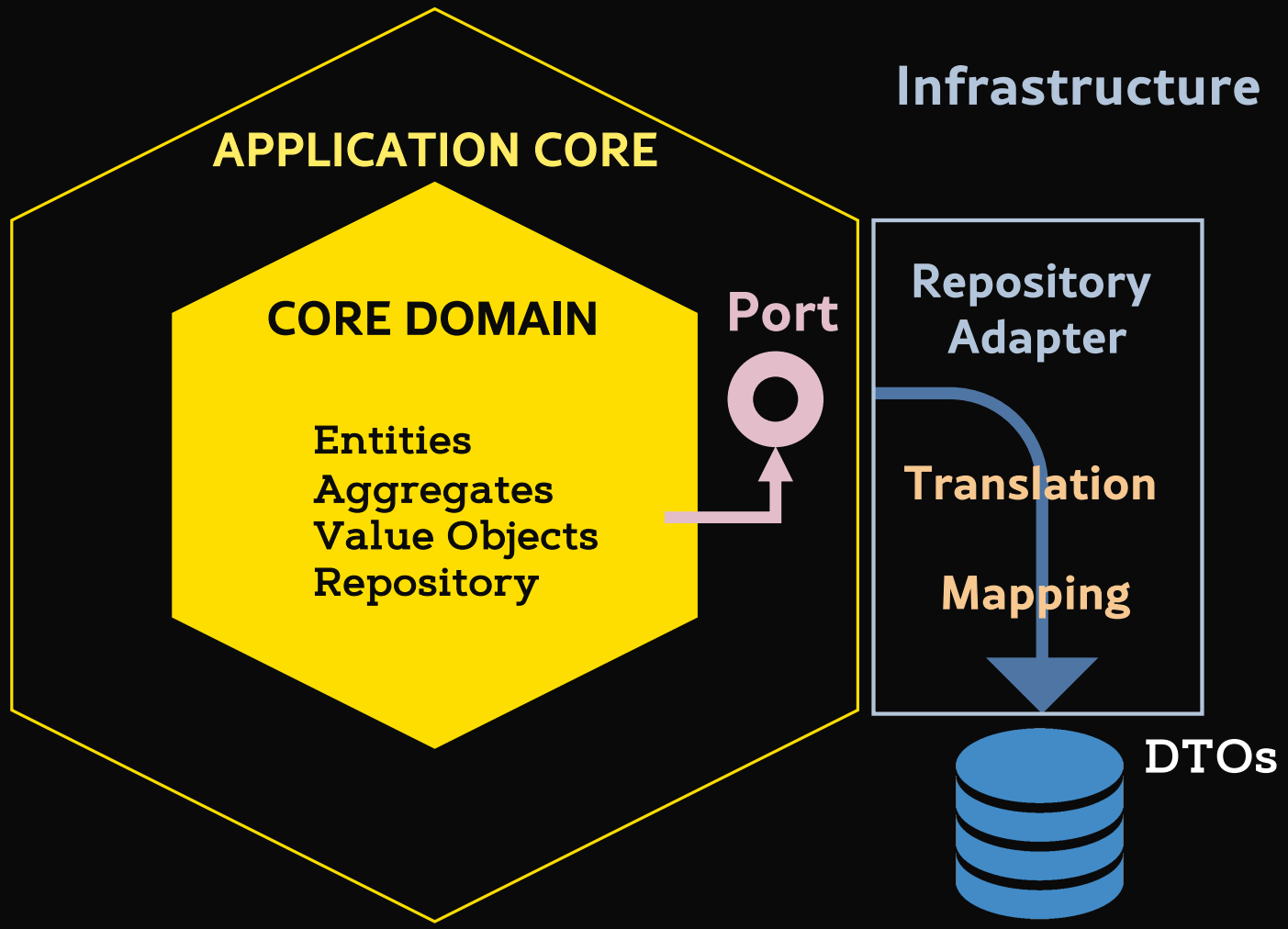
Account
SavingsGoal
PhoneNumber
Role
Transaction
Transaction

UserProfile
GoalRepository
UserRepository

Infrastructure



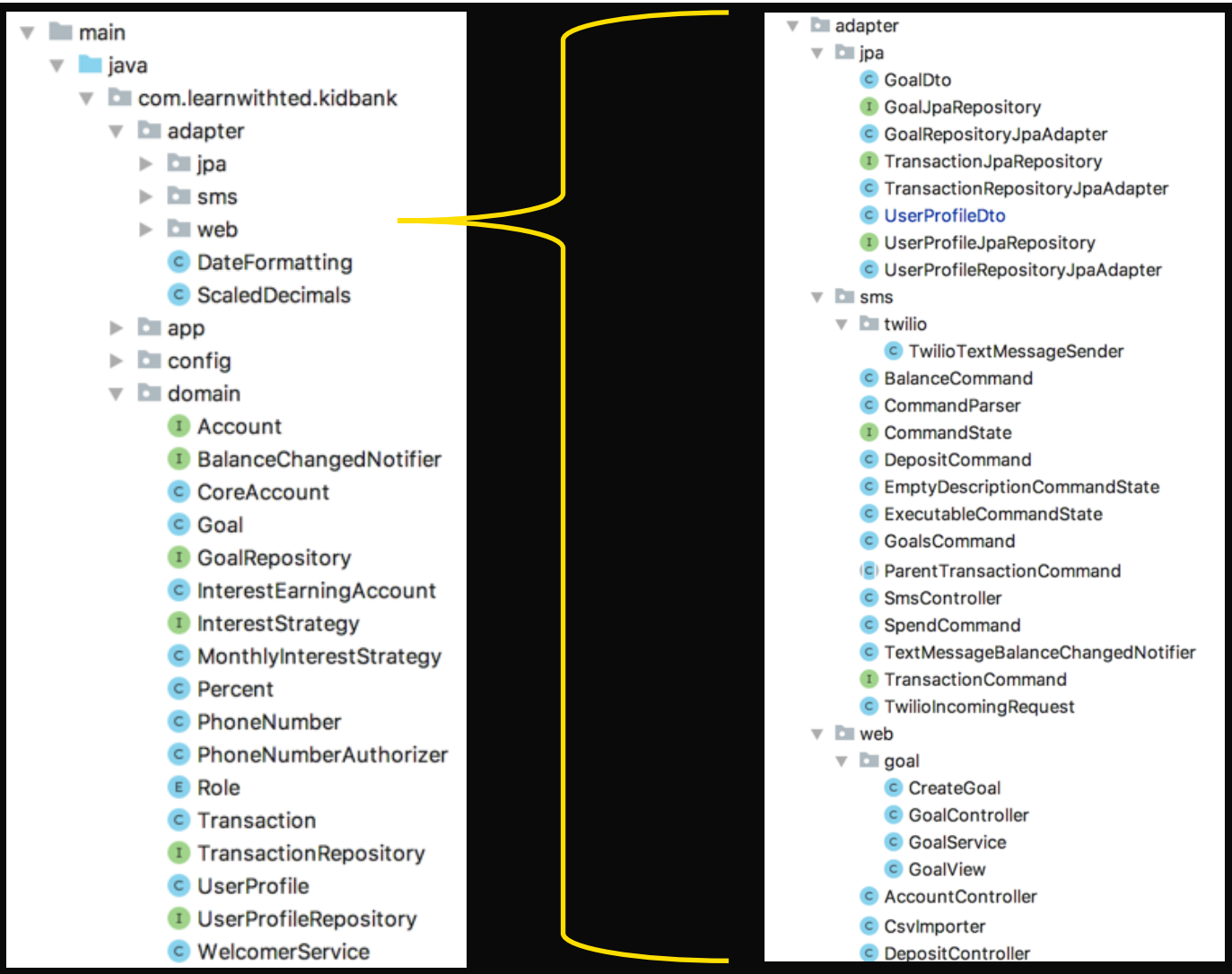

Infrastructure
CORE DOMAIN
Entities
Aggregates
Value Objects
Repository
Port
Repository
Adapter
FAKE


Infrastructure
APPLICATION CORE
CORE DOMAIN
Entities
Aggregates
Value Objects
Repository
Port
Repository
Adapter
Translation
Mapping
DTOs

Rule

Package Based on Architecture

Ted M. Young https://MoreTestable.com



@JITTERTED

Ted M. Young

https://MoreTestable.com

@JitterTed

Enforce Package Structure

ArchUnit

Ted M. Young https://MoreTestable.com

Don't Use Hexagonal

Basic CRUD

Simple or non-existent domain

Transformation/Integration-only

Hexagonal Architecture

- Focus on Core Domain
- Domain Objects Never Leave
- Adapters per Trigger Type
- Ports for Ideal Interface to Outside
- Dependencies point inwards
- Package Based on Architecture

Thank You!

Ted M. Young

Java Trainer/Coach

Spiral Learning, LLC
ted@tedmyoung.com
Live Coding: https://twitch.tv/jitterted

@JitterTed
https://ted.dev
ted.dev/youtube

@JitterTed

Ted M. Young https://MoreTestable.com