

PRODUCT MANAGEMENT

Andy Acs

PDM8

- **How the Internet Works**
- **Programming Languages & Terminology**
- **Constraints & Solutions**
- **Bringing It All Home**

- › **Define key terminology regarding creating digital products**
- › **Recognize common constraints and solutions that product managers face with digital products.**

YOU DON'T NEED TO BE AN ENGINEER, BUT YOU NEED TO SPEAK IT.

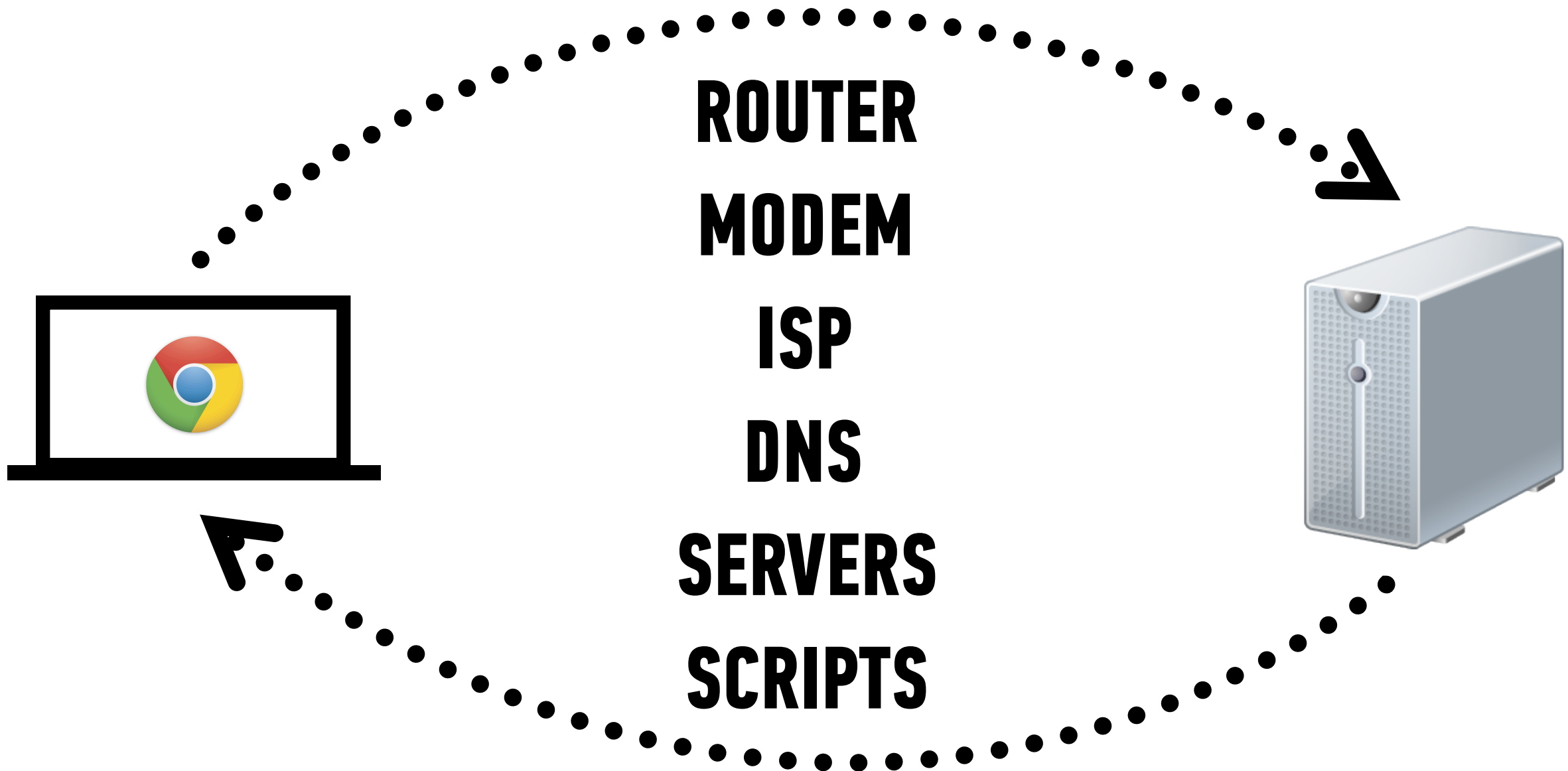
- **Having a foundational understanding of how products are built is key to:**
 - **Creating something that can realistically be built**
 - **Building a relationship with engineers**
 - **Understanding the implications of the decisions you make**

**YOU DON' T NEED TO BE AN ENGINEER,
BUT YOU NEED TO **SPEAK IT.****

**THIS LESSON IS JUST THE BEGINNING.
YOU WILL NOT LEARN EVERYTHING YOU
NEED TO KNOW BY THE END OF THIS
LESSON.**



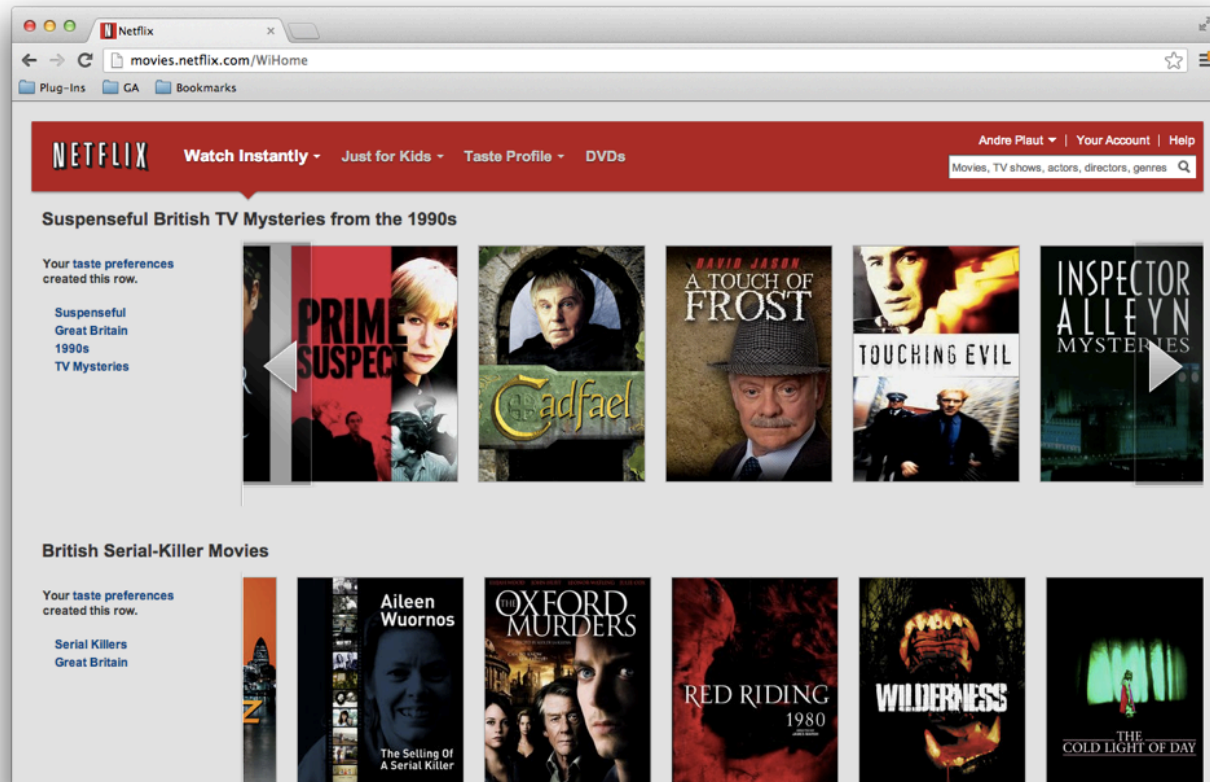
**HOW DOES
THE
INTERNET
WORK?**



**WHO HERE HAS SOME
PROGRAMMING
EXPERIENCE?**

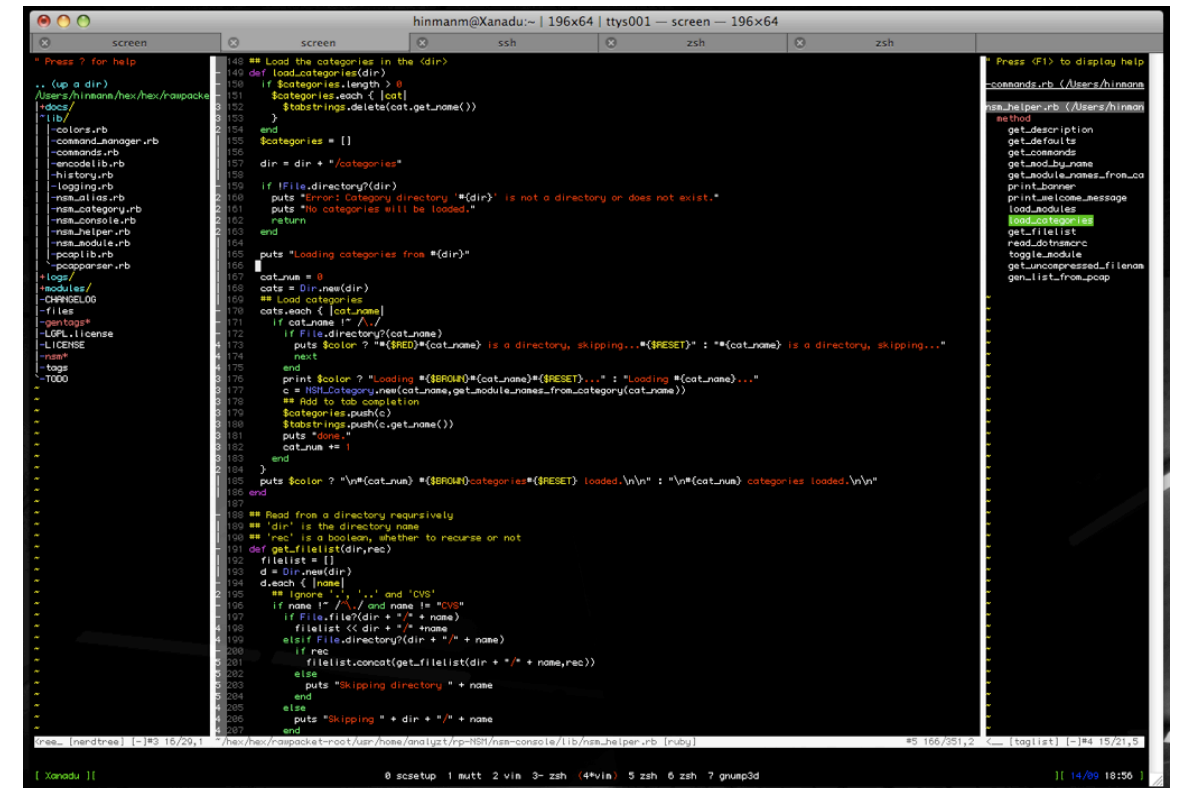
FRONT-END

WHAT USERS SEE AND INTERACT WITH



BACK-END

THE PROCESSES THAT ALLOW THE FRONT-END TO REACT, ADAPT, AND FUNCTION



PROGRAMMING LANGUAGES

FRONT-END WEB DEVELOPMENT

HTML: STRUCTURE & CONTENT (WHAT'S ON THE PAGE?);

CSS: STYLING (HOW DOES IT LOOK?);

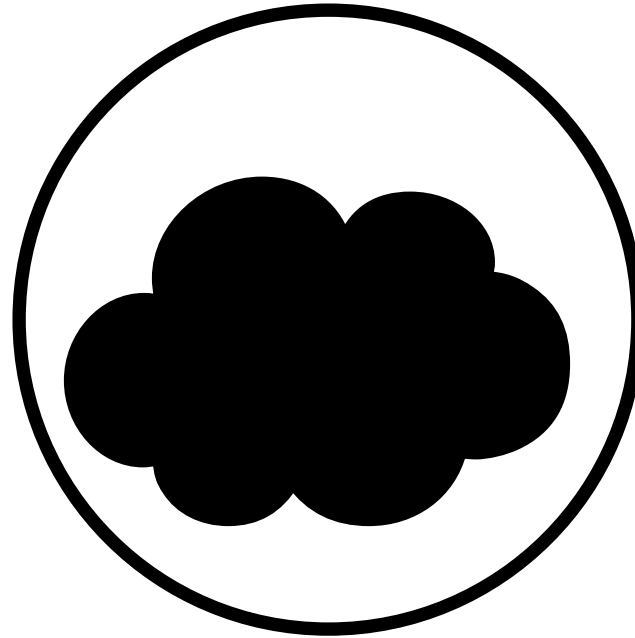
JAVASCRIPT: RESPONSE & USER ACTIVITY (HOW DO YOU INTERACT?);

BACK-END WEB DEVELOPMENT

RUBY ON RAILS

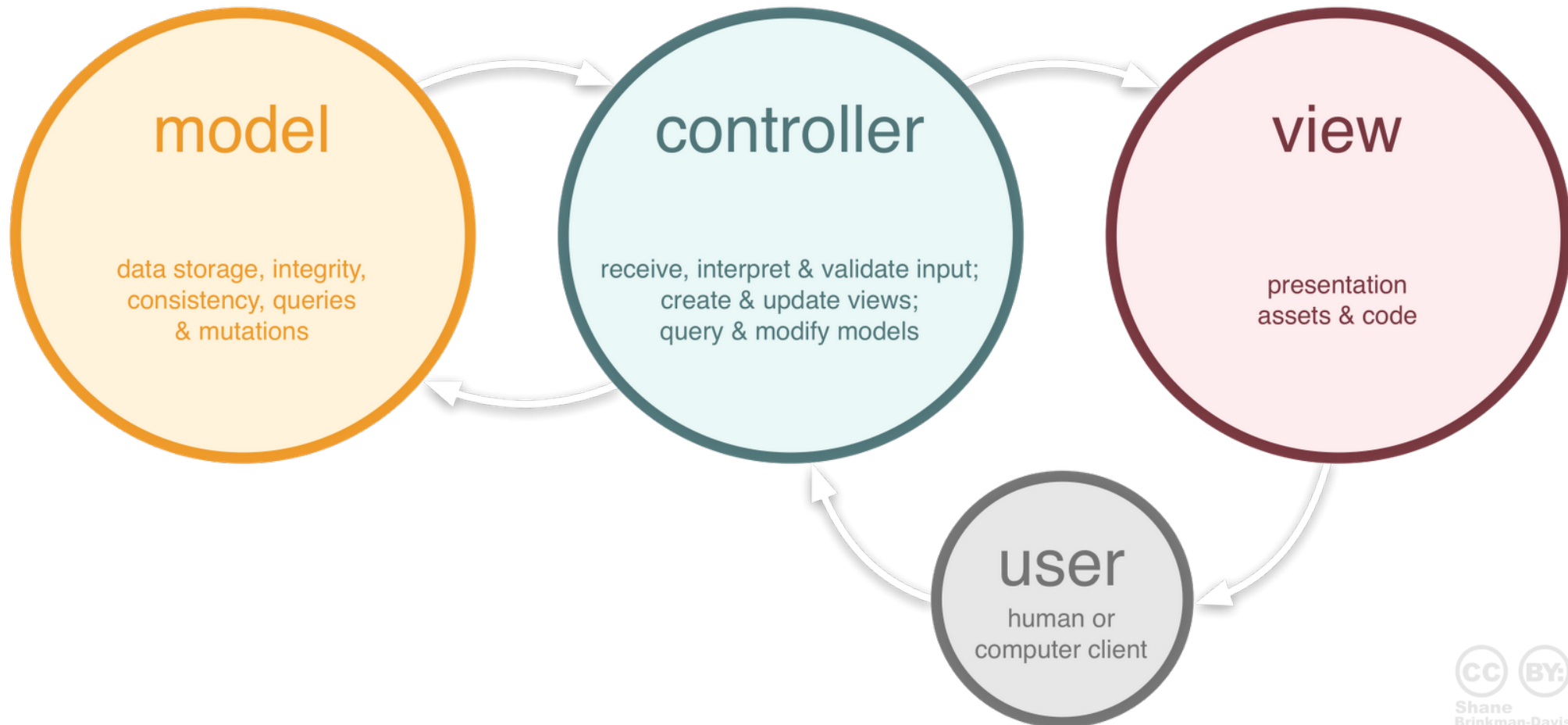
PYTHON

PHP



LET'S TAKE A BREAK.
5 MINUTES

MODEL-VIEW-CONTROLLER



MODEL

CORE BUSINESS LOGIC

WRITTEN IN RUBY, PYTHON, JAVA, C#, C++, ETC.

VIEW

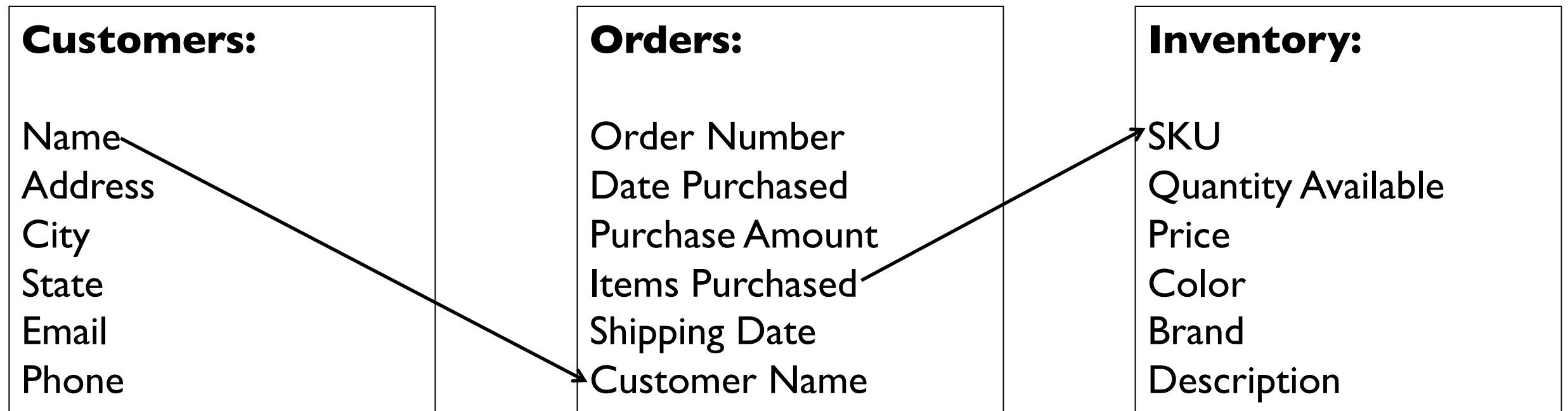
**USER INTERFACE- ALL OF YOUR ASSETS AND STYLE
HTML, CSS, JAVASCRIPT, ETC.**

CONTROLLER

**CONNECTS THE “FRONT END” AND “BACKEND”
SAME LANGUAGE AS MODEL**

HOW DOES A DATABASE ACTUALLY WORK?

What data does an e-commerce site need?



CREATE

ADDS A NEW RECORD IN THE DATABASE

READ

RETRIEVES A RECORD FROM THE DATABASE

UPDATE

CHANGES A RECORD IN THE DATABASE

DELETE

DELETES A RECORD IN THE DATABASE



**WHAT IS AN
API
ANYWAY?**



HTTP://



YOUR APP

POST

ADDS THE ENCLOSED DATA TO AN EXISTING RESOURCE

PUBLISH THIS PHOTO TO THE USERS ACCOUNT

GET

RETRIEVES A RESOURCE

GET ALL OF THE PHOTOS FOR THE SPECIFIED USER

PUT

UPDATES THE SPECIFIED RESOURCE

CHANGE THE SPECIFIED PHOTO FOR THIS USER

DELETE

DELETES THE SPECIFIED RESOURCE

DELETES THE SPECIFIED PHOTO

**SO WHAT DO YOU DO WITH ALL OF THIS
INFORMATION?**

**WORK WITH YOUR
DEVELOPERS!**

WORKING WITH DEVELOPERS

- **Should be brought in as early in the project as possible & have a sense of ownership.**
- **Present the problem you're trying to solve & solicit possible solutions.**
- **As project progresses, develop requirements that the solution needs to address, but avoid prescribing how to do it.**
- **Understand & remember that the developer is the expert in this area.**

BUILD

Do we have the in-house capability and expertise?

Is this part of our distinct competitive advantage?

Is the solution proprietary?

Can we maintain and improve this long term?

Is there no existing solution that satisfies our needs?

Can we build this cheaper or faster than an alternative?

BUY

Is there a known and trusted solution in the market?

Does this NOT play to our strengths or use our core capabilities?

Will building this distract from our core focus or prevent us from delivering more meaningful features?

Will we get to market significantly faster or less expensive?

Is it too risky to build?

PARTNER

Is there a trustworthy partner in the market?

Does their technology integrate with ours?

Does the partnership constitute a proprietary advantage?

Are we at high risk is the agreement is dissolved or the partner goes out of the business?

Are we good at technical partnerships?

BUILD

or

BUY/PARTNER

Payments

Maps/Location

CRM

Business Data

A/B Testing

Square

Google Maps

Salesforce

Foursquare

Optimizely