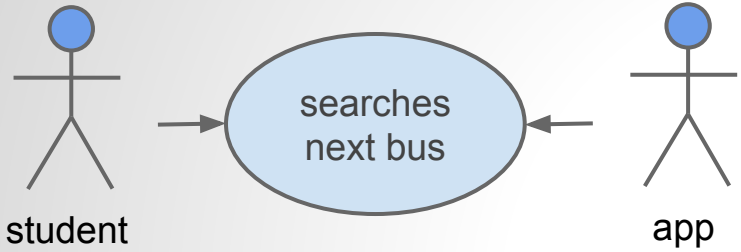
CMU Bus Tracker

Ryler Hockenbury (rhockenb)
Evgeny Toropov (etoropov)
Daniel Stoll (dstoll)

Goal - convenient bus times lookup

- simple interface
- favorite stops can be stored
- works off-line
- combines CMU shuttle and City bus schedules
- *in future*: real-time tracking

Use Cases

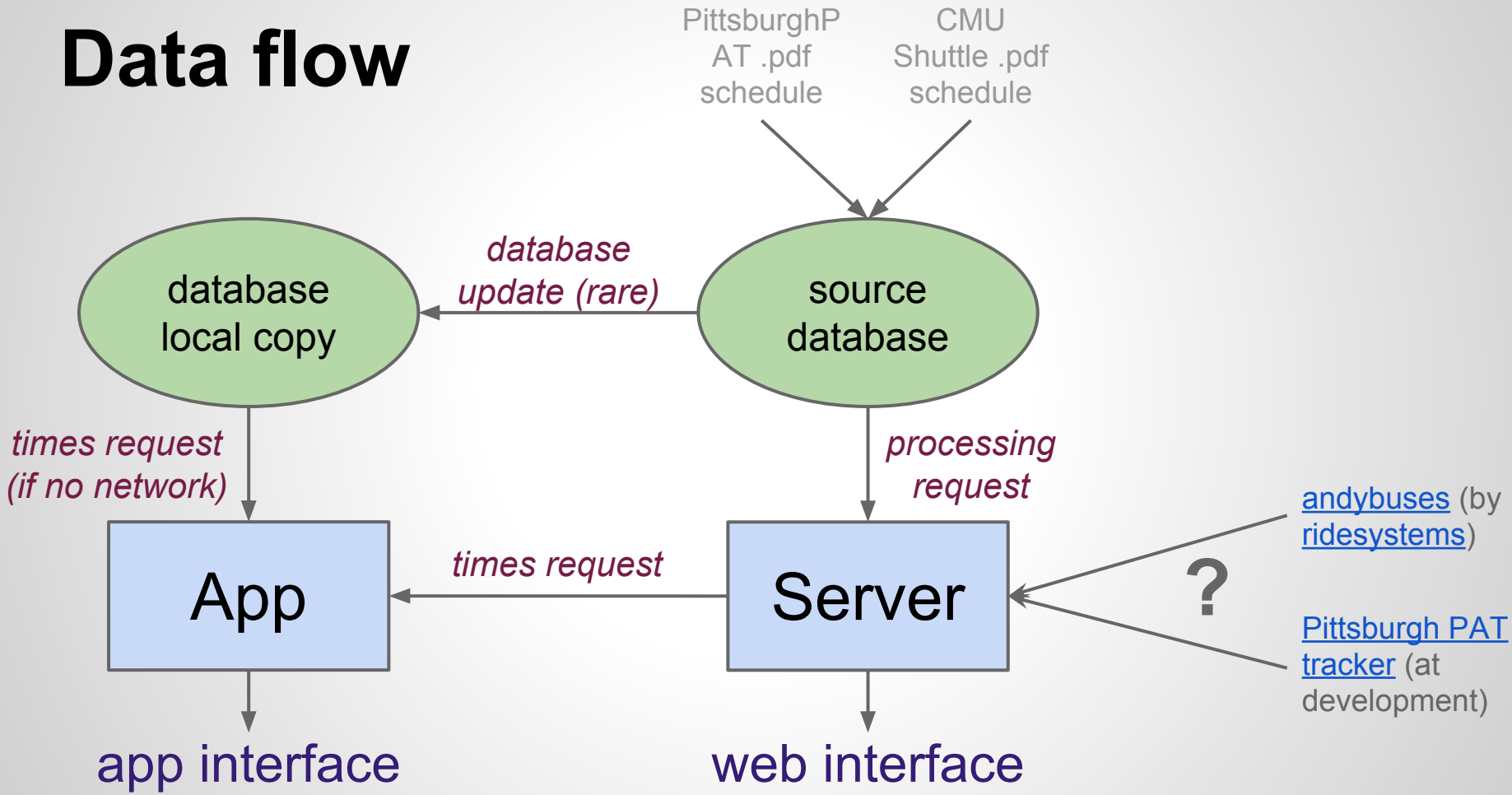

searches
next bus
student
app


customi-
zes app
student
app


Updates
schedule
on phone
student
app

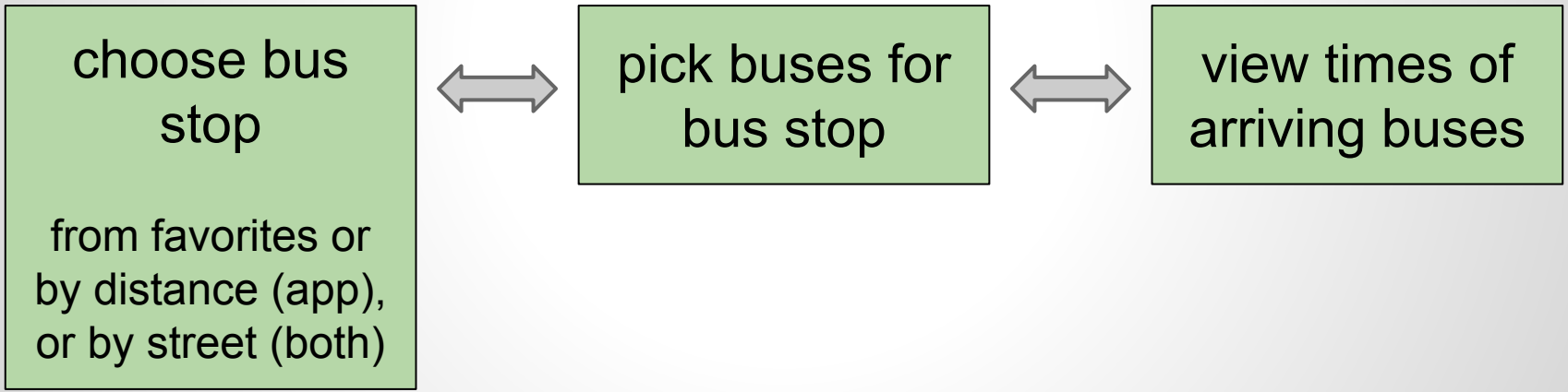

searches
next bus
student
web
interface

Data flow


PittsburghPAT .pdf schedule
CMU Shuttle .pdf schedule
database local copy
database update (rare)
source database
times request (if no network)
processing request
App
times request
Server
?
andybuses (by ridesystems)
Pittsburgh PAT tracker (at development)
app interface
web interface

Pipeline

There is App and Web interface with the same functionality



App Demo (on webcam)

- Home page
- Locating a stop / searching for a stop
- Favorites (starred)
- Settings
- Update local database
- Help on each activity
- Swipes / shake to refresh
- View Schedules

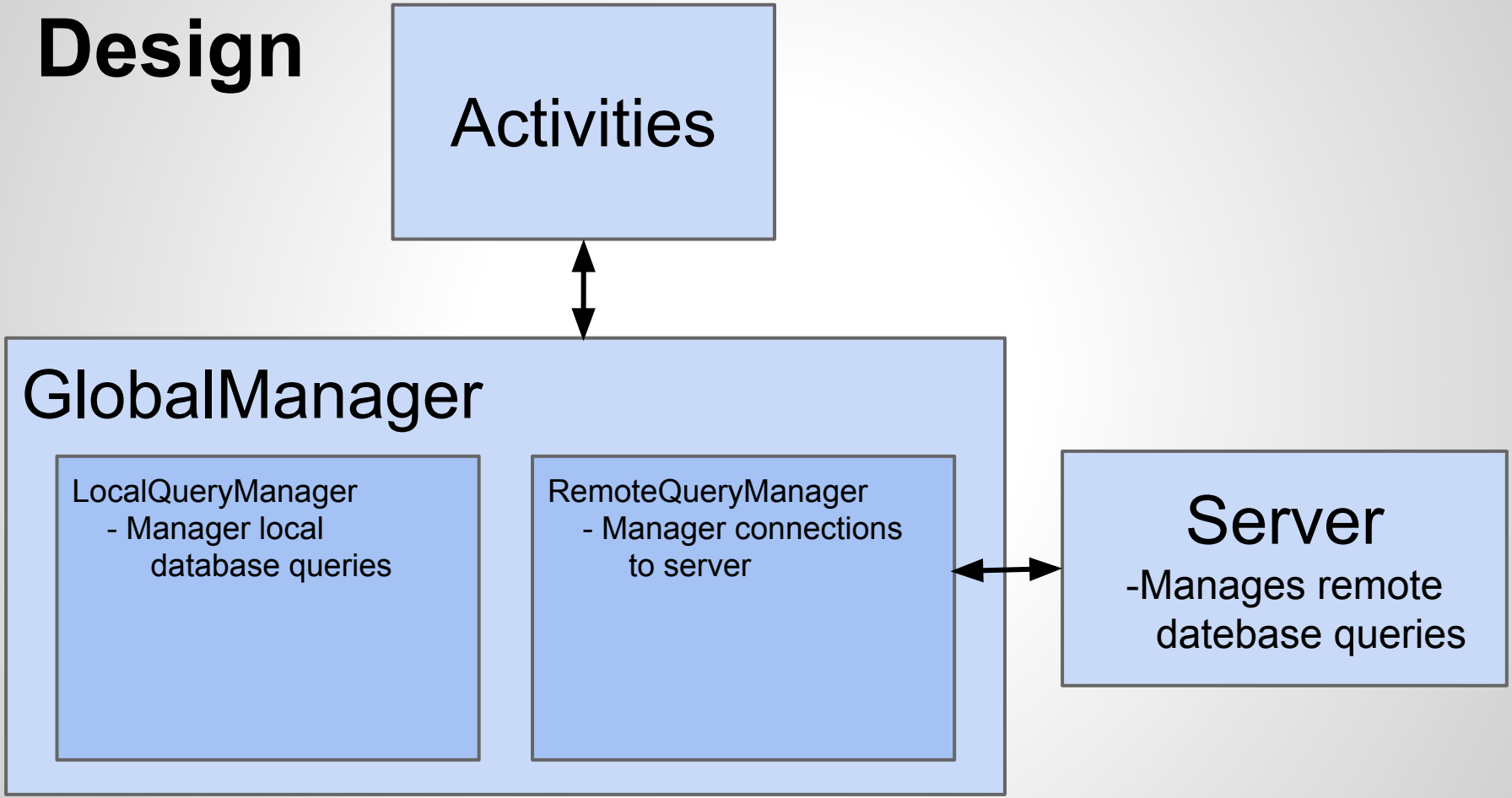
Web interface demo (screen share)

- Searching for a stop
- Picking buses
- Viewing schedule

Features

1. Hardware Audio
2. Location
3. Network-based Geo location
4. GPS
5. Accelerometer Sensor
6. Motions
7. Gestures
8. Touchscreen
9. SQL Lite DB
10. Web Services

Design


Activities
GlobalManager
LocalQueryManager
- Manager local database queries
RemoteQueryManager
- Manager connections to server
Server
-Manages remote datebase queries

Competitors

Tiramisu

real-time tracking

minimal pipeline

favorite stops can be stored

works off-line

both CMU and City schedules

is the bus full or not

andysbuses (cmu shuttle)

real-time tracking

minimal pipeline

favorite stops can be stored

works off-line

both CMU and City schedules

Google maps

real-time tracking

minimal pipeline

favorite stops can be stored

works off-line

both CMU and City schedules

“The Winter is coming”

The Game of Thrones

image from post-gazette

Questions?

Back up Slides

The Business Model Canvas

Designed for: Bus Tracker - CS Labs

Designed by: Daniel Stoll, Ryler Hockenbury, Evgeny Toropov

On: 29/10/2013

Iteration: 1

Key Partners

Who are our Key Partners?
Who are our key suppliers?
Which Key Resources are we acquiring from partners?
Which Key Activities do partners perform?

MOTIVATIONS FOR PARTNERSHIPS:
Optimization and economy
Reduction of risk and uncertainty
Acquisition of particular resources and activities

The Pittsburgh Port Authority for maps and schedules of bus routes.

Local universities advertising to their students.

Key Activities

What Key Activities do our Value Propositions require?
Our Distribution Channels?
Customer Relationships?
Revenue streams?

CATEGORIES
Production
Problem Solving
Platform/Network

Pulling updated bus schedules from the server

Supplying up to date information fro the end users

Key Resources

What Key Resources do our Value Propositions require?
Our Distribution Channels? Customer Relationships?
Revenue Streams?

TYPES OF RESOURCES
Physical
Intellectual (brand patents, copyrights, data)
Human
Financial

Up to date information from servers

Customers' faith in product

Value Propositions

What value do we deliver to the customer?
Which one of our customer's problems are we helping to solve?
What bundles of products and services are we offering to each Customer Segment?
Which customer needs are we satisfying?

CHARACTERISTICS
Newness
Performance
Customization
"Getting the Job Done"
Design
Brand/Status
Price
Cost Reduction
Risk Reduction
Accessibility
Convenience/Usability

Allows user to save time when traveling by bus

Saves a lot of frustration brought about by late buses and hard to find schedules

Customer Relationships

What type of relationship does each of our Customer Segments expect us to establish and maintain with them?
Which ones have we established?
How are they integrated with the rest of our business model?
How costly are they?

EXAMPLES
Personal assistance
Dedicated Personal Assistance
Self-Service
Automated Services
Communities
Co-creation

Reliability and endurance by staying up to date with current bus schedules.

Channels

Through which Channels do our Customer Segments want to be reached?
How are we reaching them now?
How are our Channels integrated?
Which ones work best?
Which ones are most cost-efficient?
How are we integrating them with customer routines?

CHANNEL PHASES
1. *Awareness*
2. *Evaluation*
3. *Purchase*
4. *Delivery*
5. *After sales*

Advertising in app store

Information sessions on Carnegie Mellon campus

Customer Segments

For whom are we creating value?
Who are our most important customers?

Mass Market
Niche Market
Segmented
Diversified
Multi-sided Platform

Students and Professors at Carnegie Mellon and University of Pittsburgh

Cost Structure

What are the most important costs inherent in our business model?
Which Key Resources are most expensive?
Which Key Activities are most expensive?

IS YOUR BUSINESS MORE:
Cost Driven (leanest cost structure, low price value proposition, maximum automation, extensive outsourcing)
Value Driven (focused on value creation, premium value proposition)

SAMPLE CHARACTERISTICS:
Fixed Costs (salaries, rents, utilities)
Variable costs
Economies of scale
Economies of scope

Marketing, promotion and communications

Staff costs

Production of application

Revenue Streams

For what value are our customers really willing to pay?
For what do they currently pay?
How are they currently paying?
How would they prefer to pay?
How much does each Revenue Stream contribute to overall revenues?

TYPES
Asset sale
Usage fee
Subscription Fees
Lending/Renting/Leasing
Licensing
Brokerage fees
Advertising

FIXED PRICING
List Price
Product feature dependent
Customer segment dependent
Volume dependent

DYNAMIC PRICING
Negotiation(bargaining)
Yield Management
Real-time-Market

Purchases made through app store

Expectation from customers will be low pricing

Testing

- Activity testing (intents, swipes, mediaplayer sounds, motions)
- Database testing (queries return correct data)
- Server testing (returning populated data structures)
- Use Case testing