Lecture 3:
Goal-oriented interaction

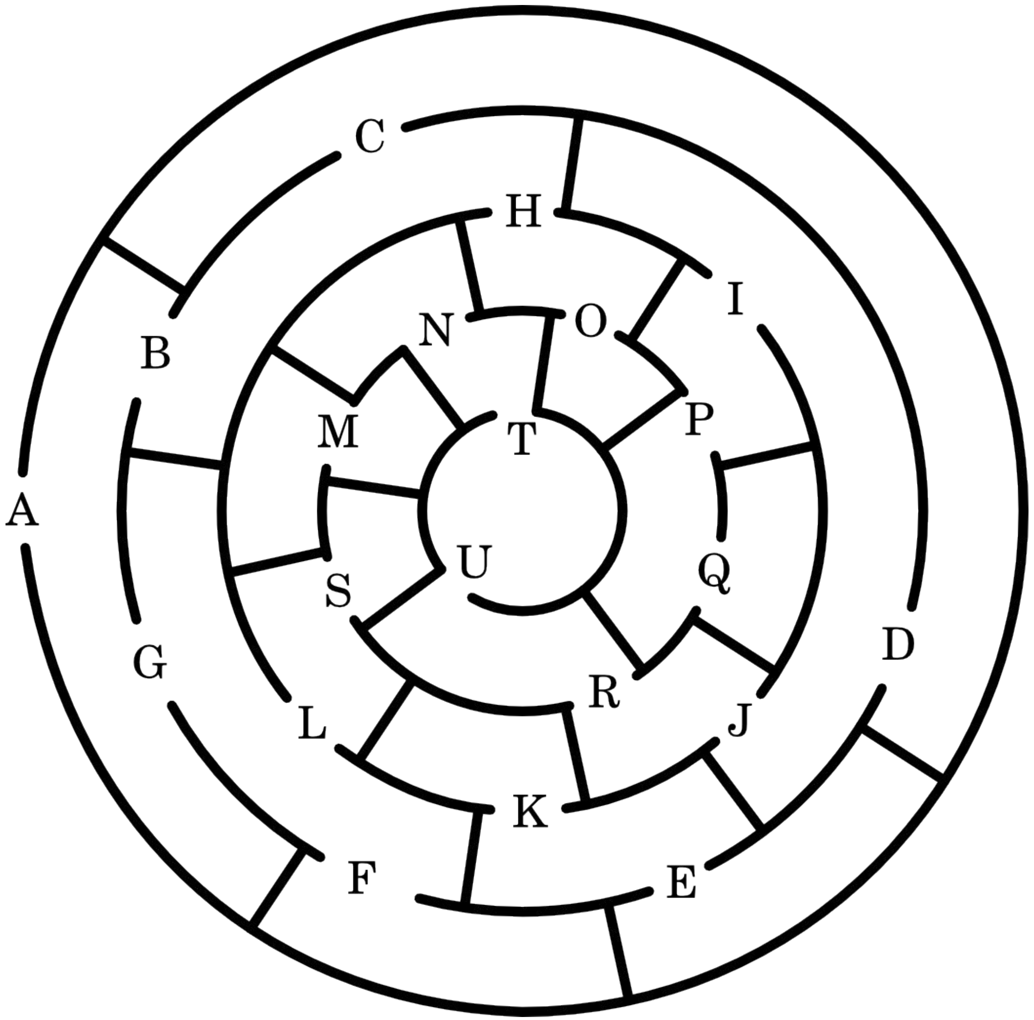
Using cognitive theories of planning, learning and understanding to understand user behaviour, and what they find hard.

Overview of the course

- Theory driven approaches to HCI
- Design of visual displays
- **Goal-oriented interaction**
- Designing efficient systems
- Designing smart systems (guest lecturer)
- Designing meaningful systems (guest lecturer)
- Evaluating interactive system designs
- Designing complex systems

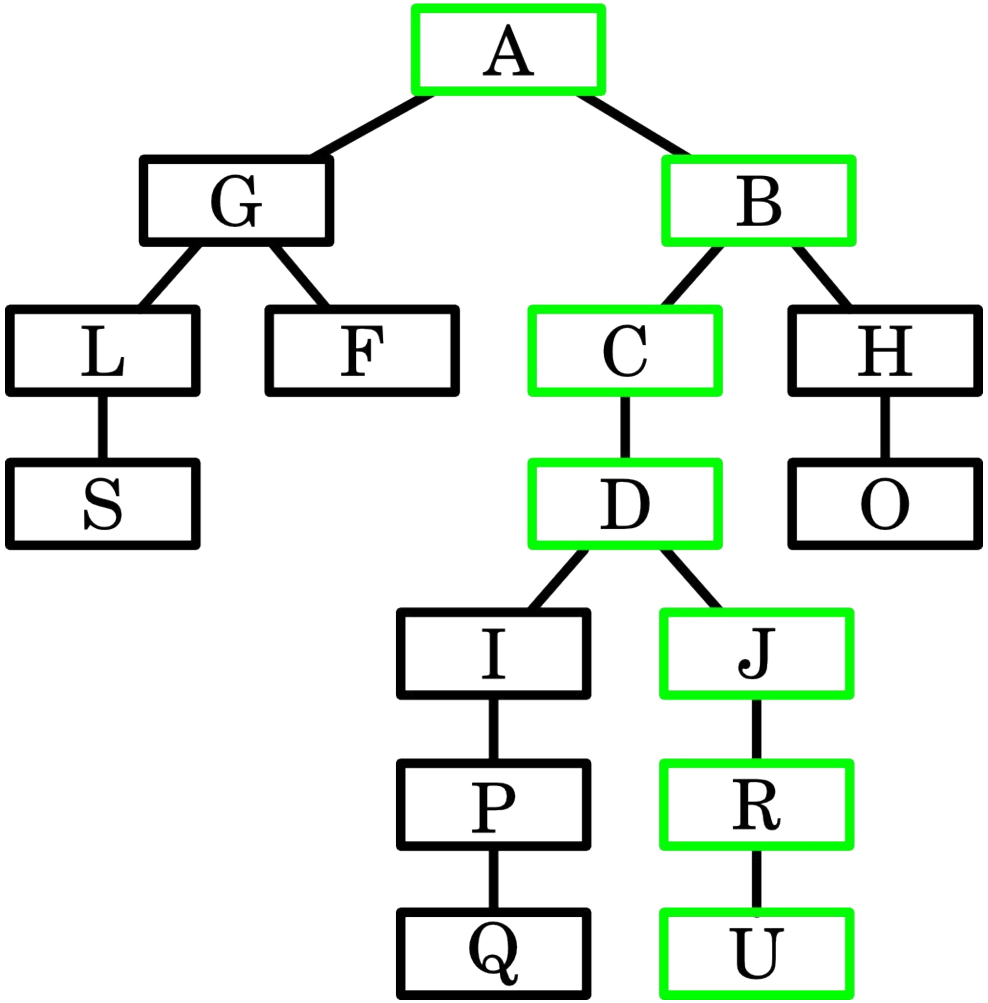
A *Metatheory* (in first-wave HCI): User interaction can be modelled as search

To come in Prolog course (later this term): problem solving using graph search



From Rice & Beresford

Turn the problem into a graph

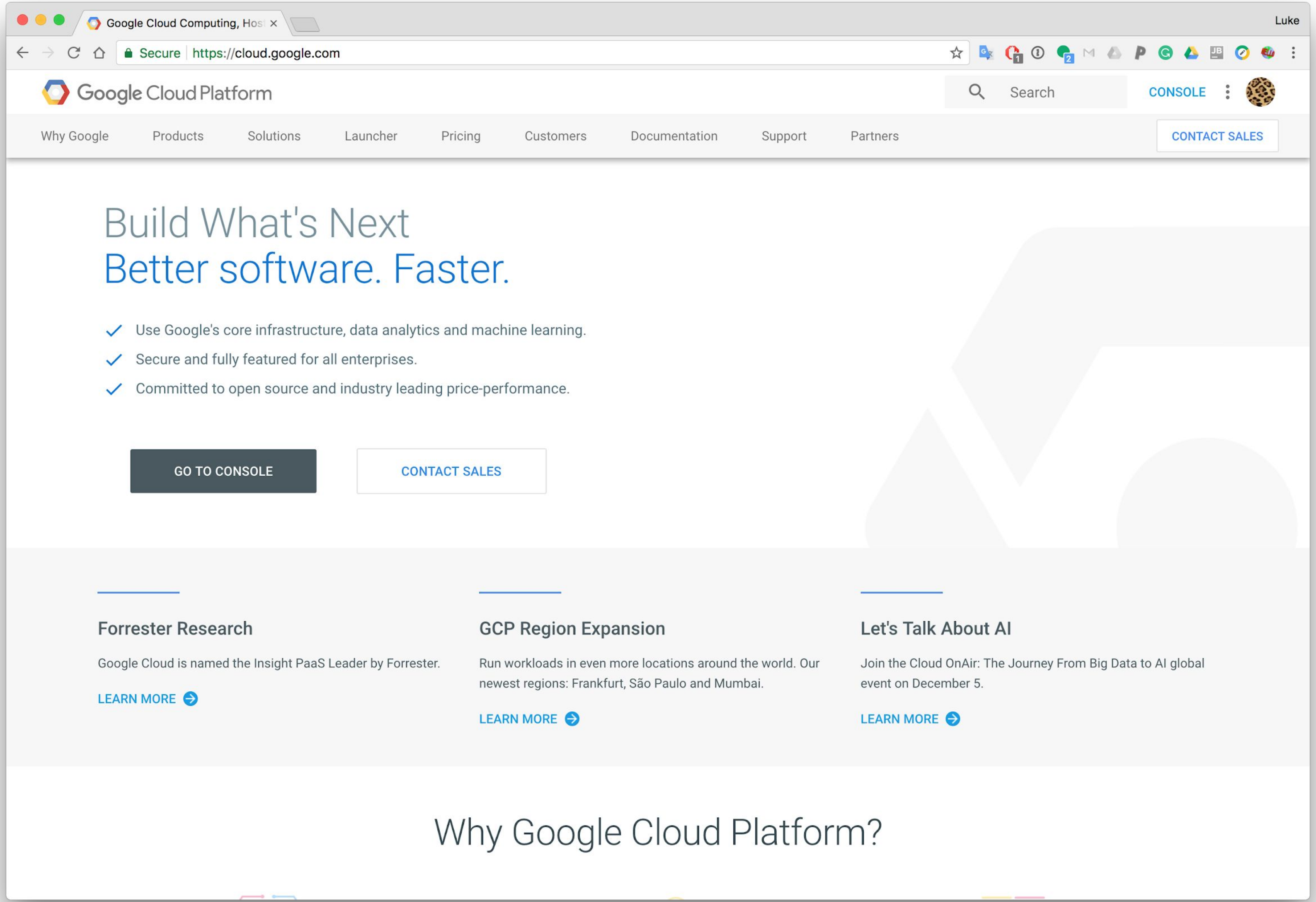


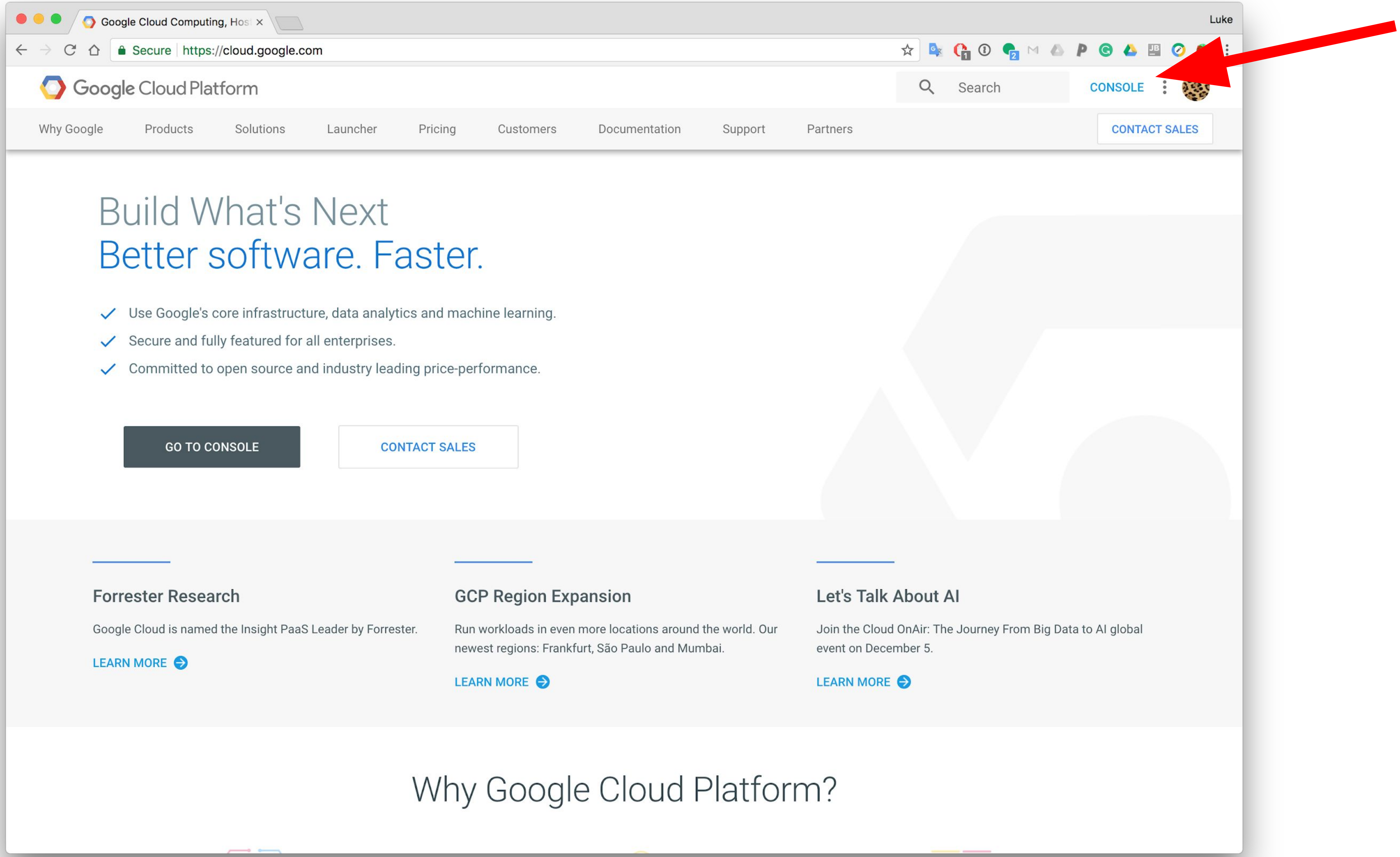
Encode as Prolog facts to solve

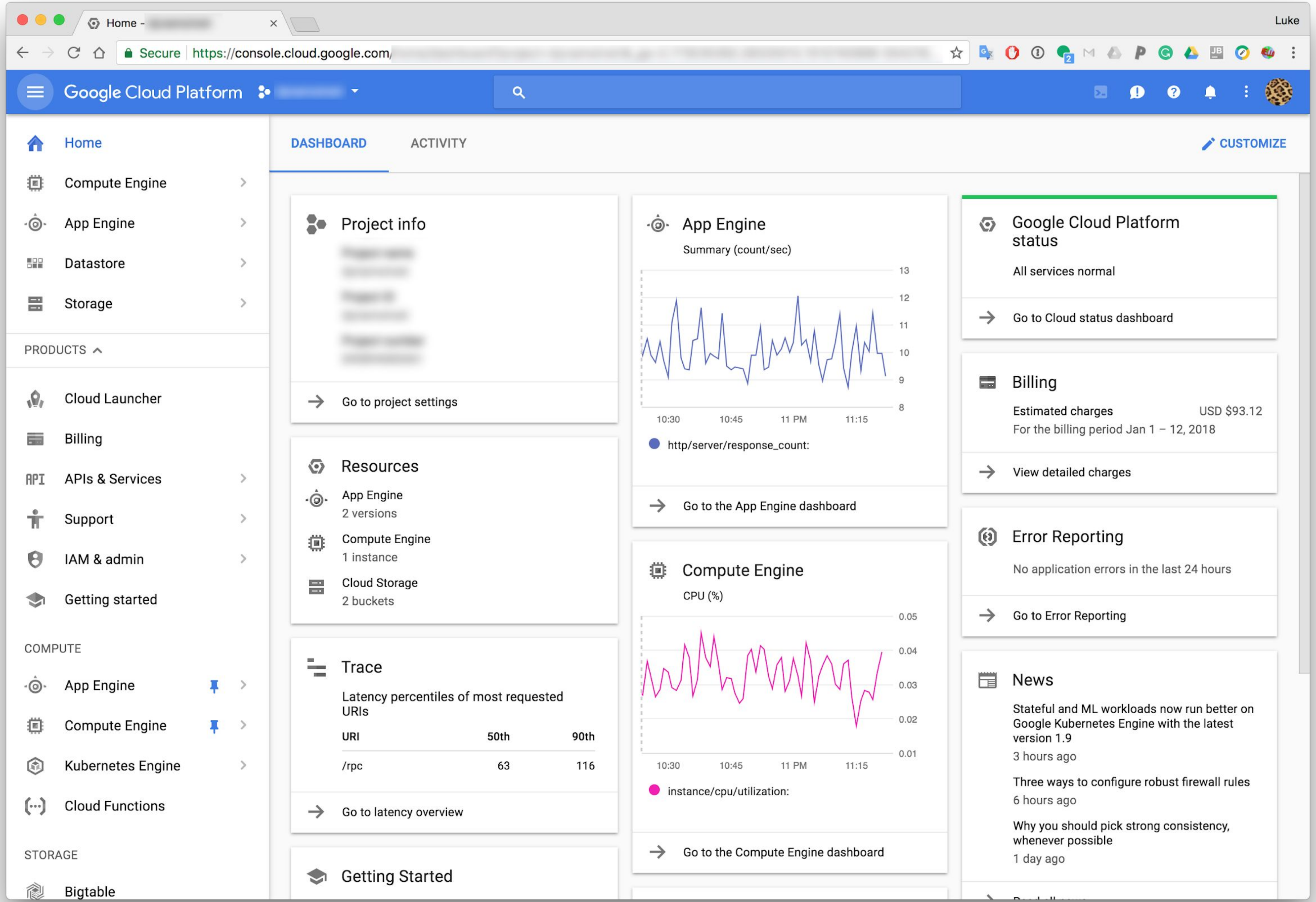
```
route(a,g).
route(g,l).
route(l,s).
...
travel(A,A).
travel(A,C) :- route(A,B),travel(B,C).

solve :- start(A),finish(B), travel(A,B).
```

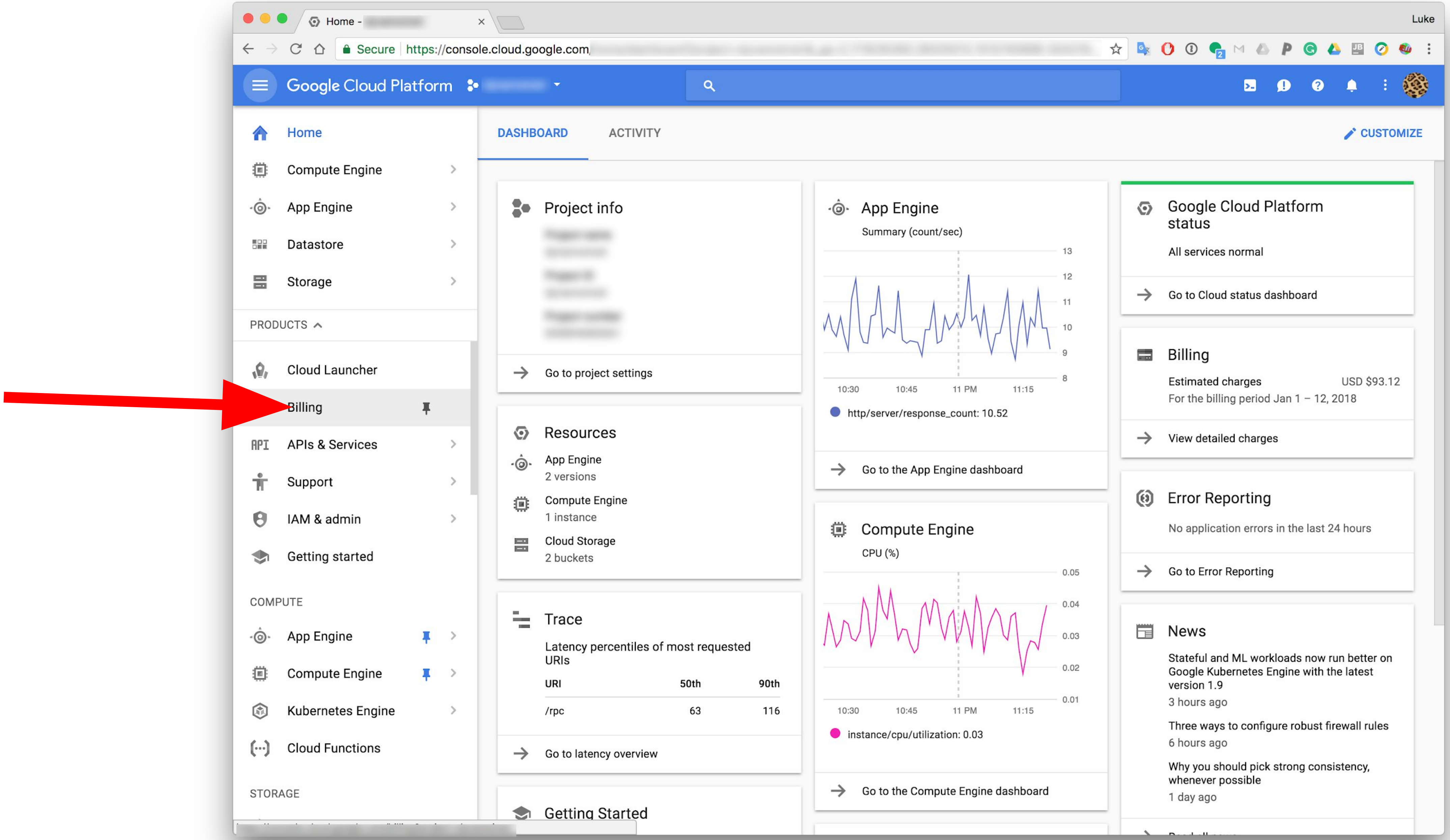
```
start(a).
finish(u).
```

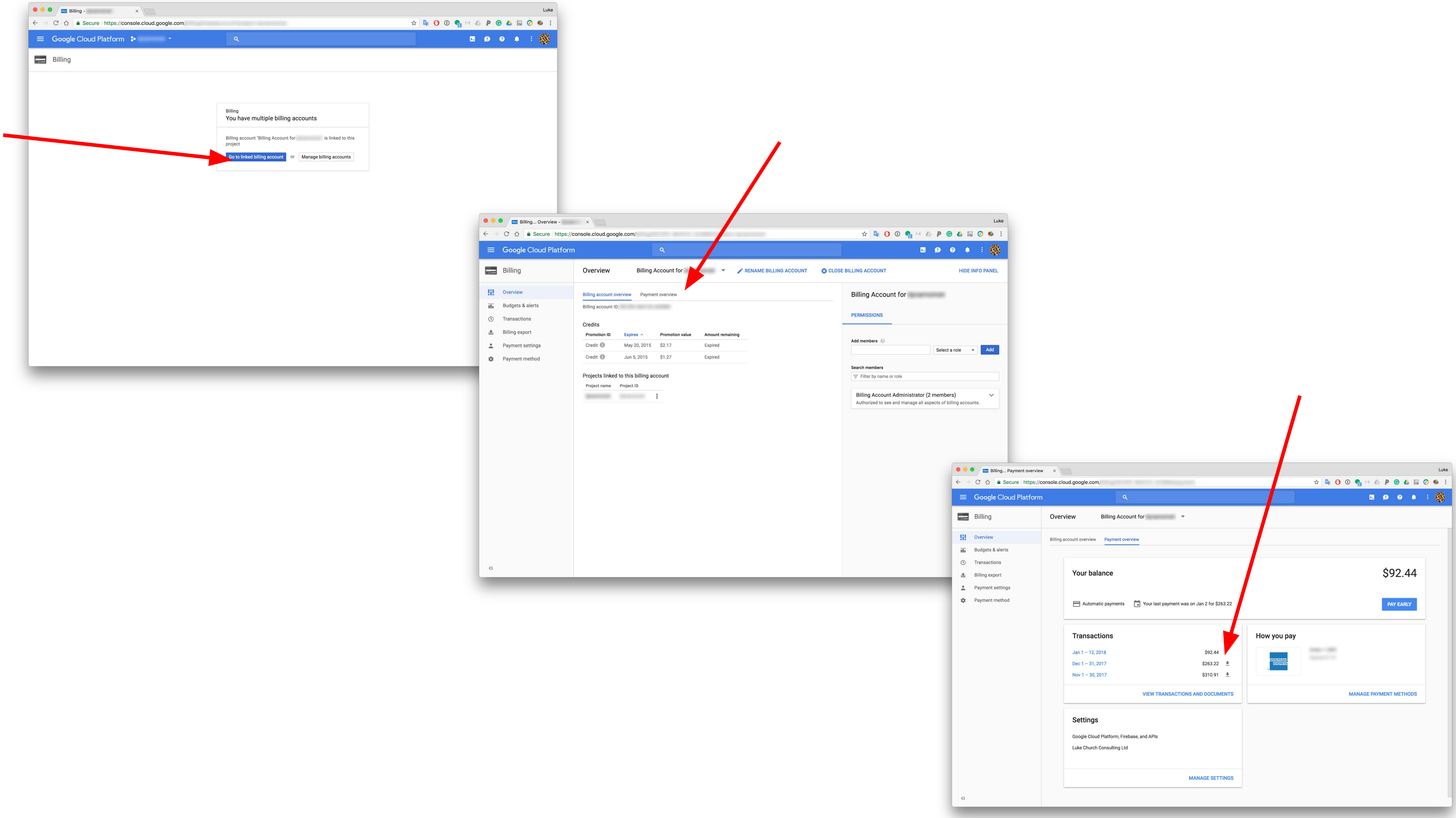
HCI example of a **User Goal**:
"How much did my use of Google Cloud Platform cost me last month?"


Google Cloud Computing, Hos
Luke
Secure
https://cloud.google.com
Google Cloud Platform
Search
CONSOLE
Why Google
Products
Solutions
Launcher
Pricing
Customers
Documentation
Support
Partners
CONTACT SALES
Build What's Next
Better software. Faster.
Use Google's core infrastructure, data analytics and machine learning.
Secure and fully featured for all enterprises.
Committed to open source and industry leading price-performance.
GO TO CONSOLE
CONTACT SALES
Forrester Research
Google Cloud is named the Insight PaaS Leader by Forrester.
LEARN MORE
GCP Region Expansion
Run workloads in even more locations around the world. Our newest regions: Frankfurt, São Paulo and Mumbai.
LEARN MORE
Let's Talk About AI
Join the Cloud OnAir: The Journey From Big Data to AI global event on December 5.
LEARN MORE
Why Google Cloud Platform?


Google Cloud Computing, Hos
Luke
Secure https://cloud.google.com
Google Cloud Platform
Search
CONSOLE
Why Google
Products
Solutions
Launcher
Pricing
Customers
Documentation
Support
Partners
CONTACT SALES
Build What's Next
Better software. Faster.
Use Google's core infrastructure, data analytics and machine learning.
Secure and fully featured for all enterprises.
Committed to open source and industry leading price-performance.
GO TO CONSOLE
CONTACT SALES
Forrester Research
Google Cloud is named the Insight PaaS Leader by Forrester.
LEARN MORE
GCP Region Expansion
Run workloads in even more locations around the world. Our newest regions: Frankfurt, São Paulo and Mumbai.
LEARN MORE
Let's Talk About AI
Join the Cloud OnAir: The Journey From Big Data to AI global event on December 5.
LEARN MORE
Why Google Cloud Platform?


Home -
Luke
Secure
https://console.cloud.google.com/
Google Cloud Platform
DASHBOARD
ACTIVITY
CUSTOMIZE
Home
Compute Engine
App Engine
Datastore
Storage
PRODUCTS
Cloud Launcher
Billing
APIs & Services
Support
IAM & admin
Getting started
COMPUTE
App Engine
Compute Engine
Kubernetes Engine
Cloud Functions
STORAGE
Bigtable
Project info
Go to project settings
Resources
App Engine
2 versions
Compute Engine
1 instance
Cloud Storage
2 buckets
Trace
Latency percentiles of most requested URIs
URI 50th 90th
/rpc 63 116
Go to latency overview
Getting Started
App Engine
Summary (count/sec)
13
12
11
10
9
8
10:30
10:45
11 PM
11:15
http/server/response_count:
Go to the App Engine dashboard
Compute Engine
CPU (%)
0.05
0.04
0.03
0.02
0.01
10:30
10:45
11 PM
11:15
instance/cpu/utilization:
Go to the Compute Engine dashboard
Google Cloud Platform status
All services normal
Go to Cloud status dashboard
Billing
Estimated charges USD $93.12
For the billing period Jan 1 – 12, 2018
View detailed charges
Error Reporting
No application errors in the last 24 hours
Go to Error Reporting
News
Stateful and ML workloads now run better on Google Kubernetes Engine with the latest version 1.9
3 hours ago
Three ways to configure robust firewall rules
6 hours ago
Why you should pick strong consistency, whenever possible
1 day ago


Home -
Luke
Secure
https://console.cloud.google.com/
Google Cloud Platform
Home
Compute Engine
App Engine
Datastore
Storage
PRODUCTS
Cloud Launcher
Billing
APIs & Services
Support
IAM & admin
Getting started
COMPUTE
App Engine
Compute Engine
Kubernetes Engine
Cloud Functions
STORAGE
DASHBOARD
ACTIVITY
CUSTOMIZE
Project info
Go to project settings
Resources
App Engine
2 versions
Compute Engine
1 instance
Cloud Storage
2 buckets
Trace
Latency percentiles of most requested URIs
URI
50th
90th
/rpc
63
116
Go to latency overview
Getting Started
App Engine
Summary (count/sec)
13
12
11
10
9
8
10:30
10:45
11 PM
11:15
http/server/response_count: 10.52
Go to the App Engine dashboard
Compute Engine
CPU (%)
0.05
0.04
0.03
0.02
0.01
10:30
10:45
11 PM
11:15
instance/cpu/utilization: 0.03
Go to the Compute Engine dashboard
Google Cloud Platform status
All services normal
Go to Cloud status dashboard
Billing
Estimated charges
USD $93.12
For the billing period Jan 1 – 12, 2018
View detailed charges
Error Reporting
No application errors in the last 24 hours
Go to Error Reporting
News
Stateful and ML workloads now run better on Google Kubernetes Engine with the latest version 1.9
3 hours ago
Three ways to configure robust firewall rules
6 hours ago
Why you should pick strong consistency, whenever possible
1 day ago

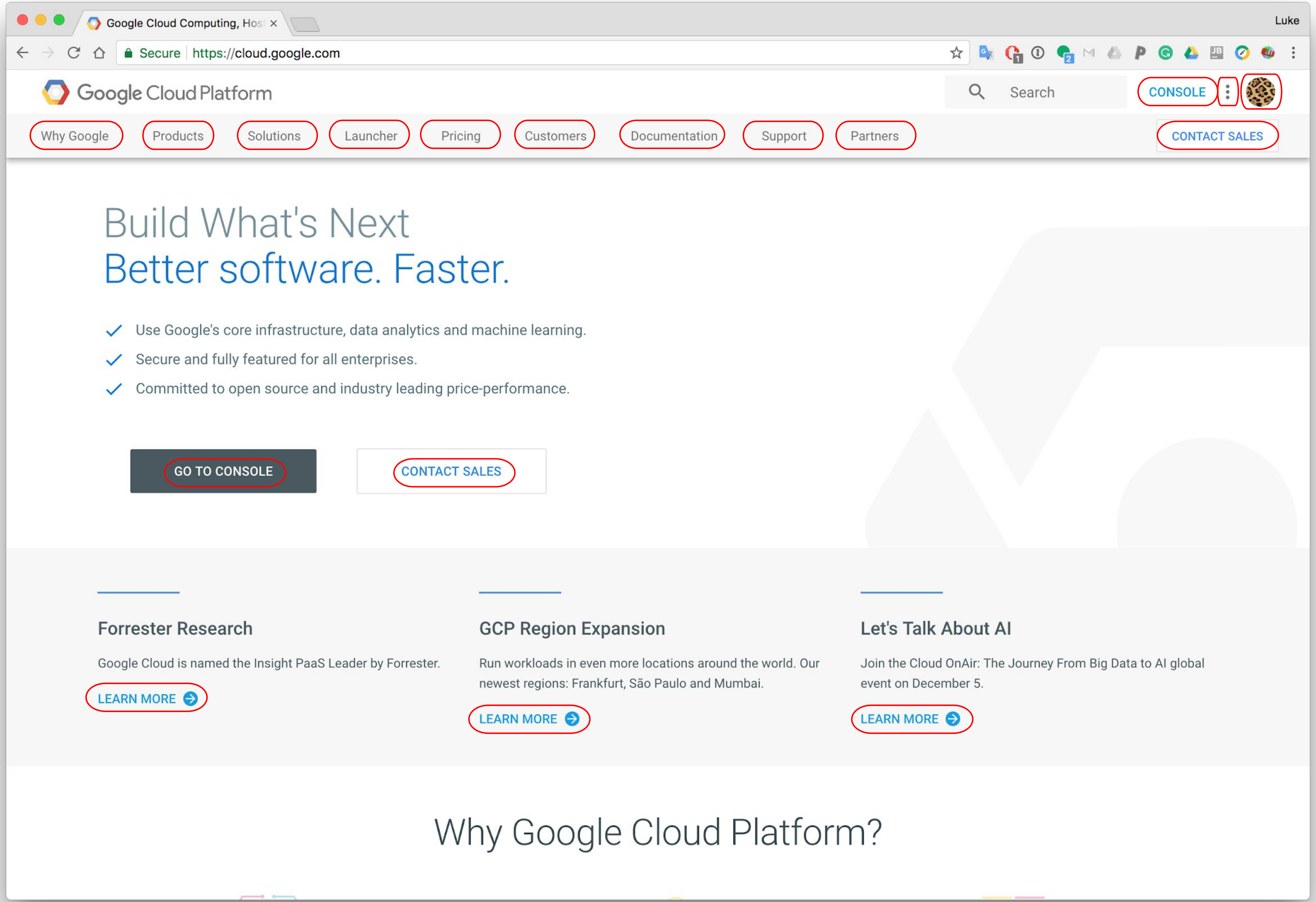


Billing
You have multiple billing accounts
Billing account "Billing Account for is linked to this project
Go to linked billing account
or
Manage billing accounts
Google Cloud Platform
Overview
Billing Account for
RENAME BILLING ACCOUNT
CLOSE BILLING ACCOUNT
HIDE INFO PANEL
Budgets & alerts
Transactions
Billing export
Payment settings
Payment method
Billing account overview
Payment overview
Credits
Promotion ID
Expires
Promotion value
Amount remaining
Credit May 20, 2015 $2.17 Expired
Credit Jun 5, 2015 $1.27 Expired
Projects linked to this billing account
Project name
Project ID
PERMISSIONS
Add members
Select a role
Add
Search members
Filter by name or role
Billing Account Administrator (2 members)
Authorized to see and manage all aspects of billing accounts.
Your balance
$92.44
Automatic payments
Your last payment was on Jan 2 for $263.22
PAY EARLY
Transactions
Jan 1 – 12, 2018 $92.44
Dec 1 – 31, 2017 $263.22
Nov 1 – 30, 2017 $310.91
VIEW TRANSACTIONS AND DOCUMENTS
How you pay
MANAGE PAYMENT METHODS
Settings
Google Cloud Platform, Firebase, and APIs
Luke Church Consulting Ltd
MANAGE SETTINGS

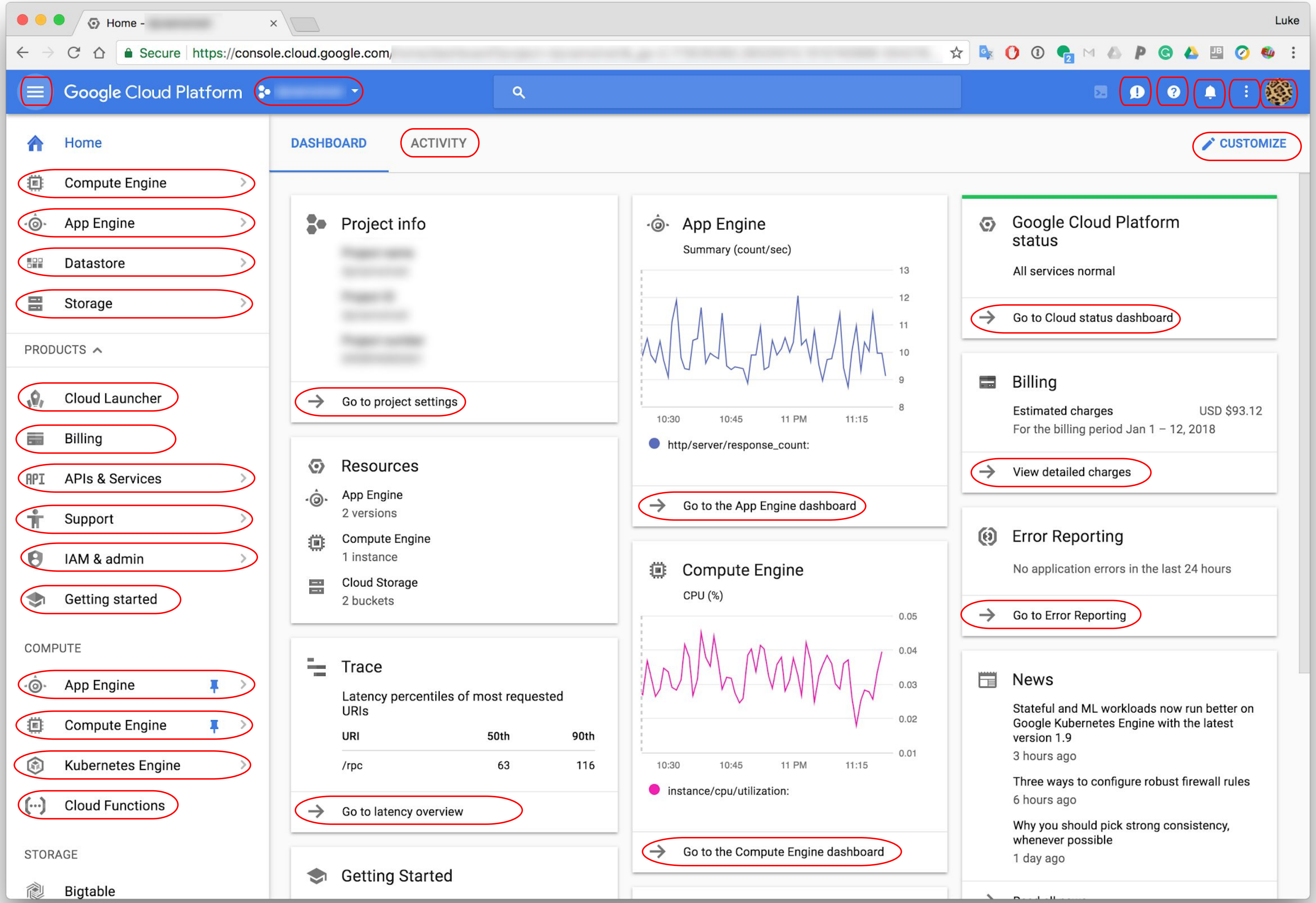
What search algorithm is being used here?

Breadth first/Depth first?

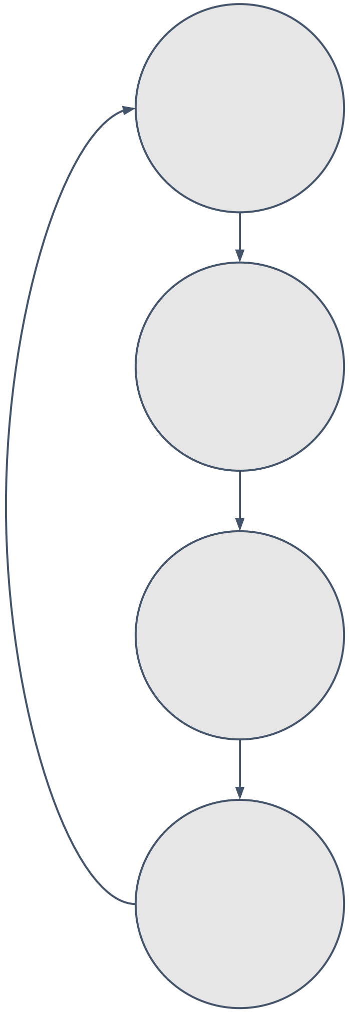
Click targets



Click targets



[Simplified] Cognitive Walkthrough



Goal

Availability

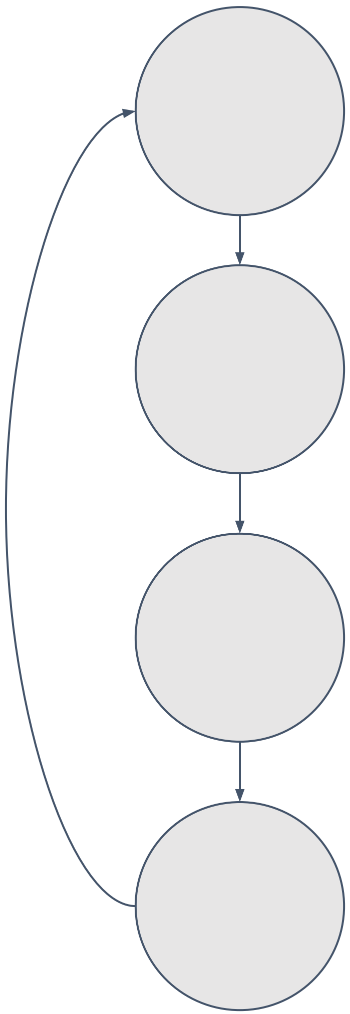
Match

Feedback

See:
https://www.colorado.edu/ics/sites/default/files/attached-files/93-07.pdf

For a detailed description

Finding your bill?

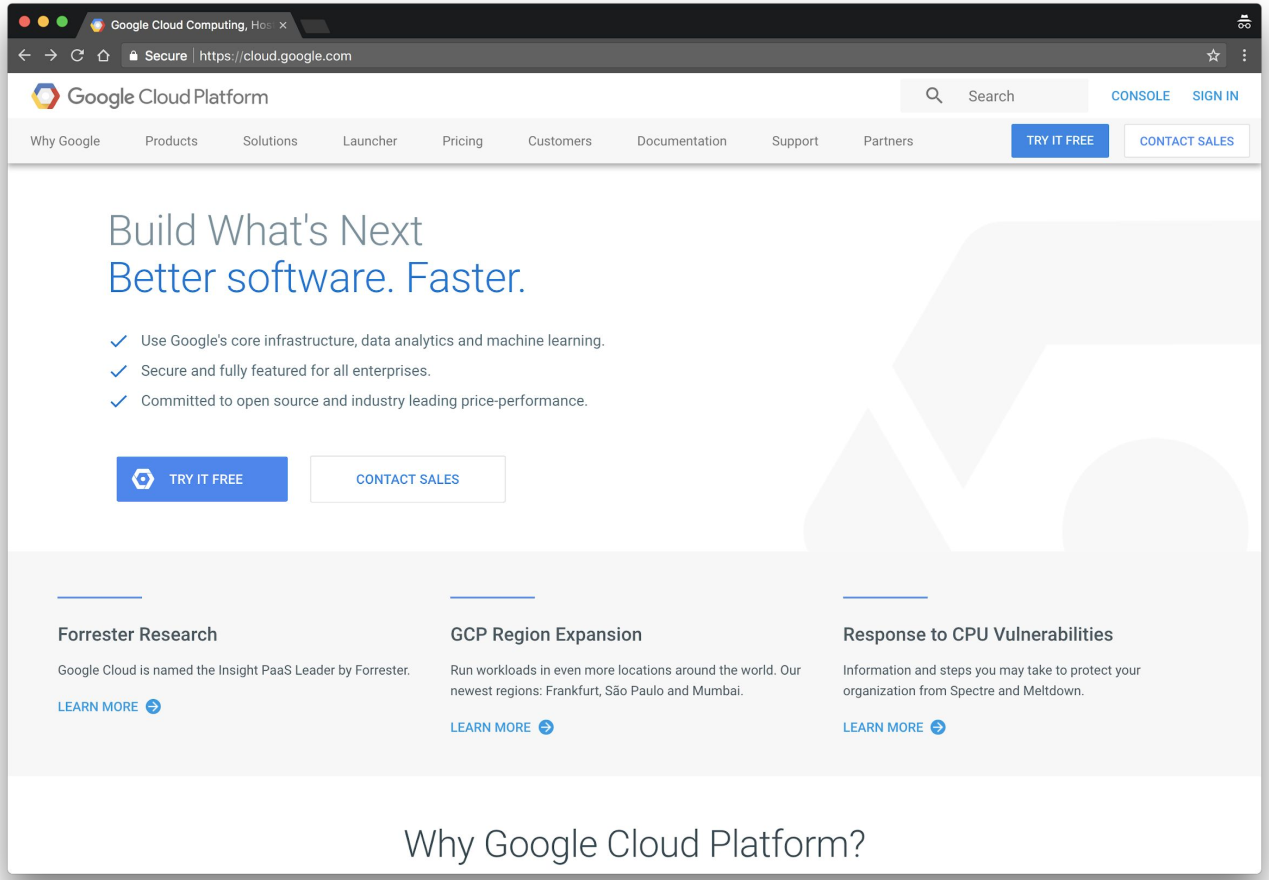


Goal

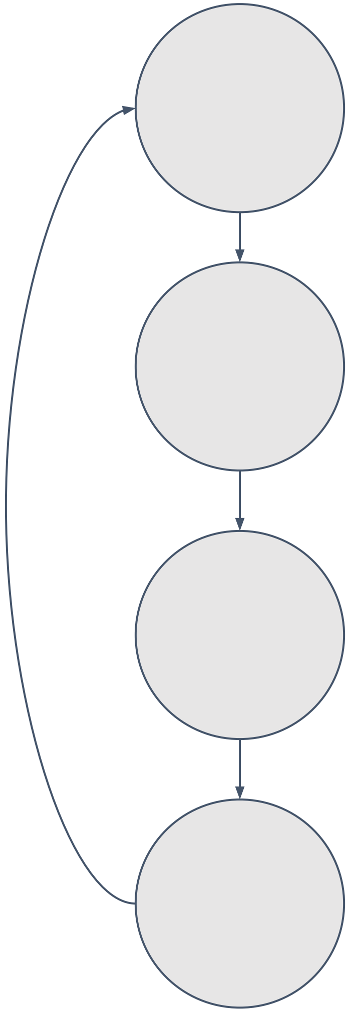
Availability

Match

Feedback



Example: Walkthrough of an API (demo)



Goal

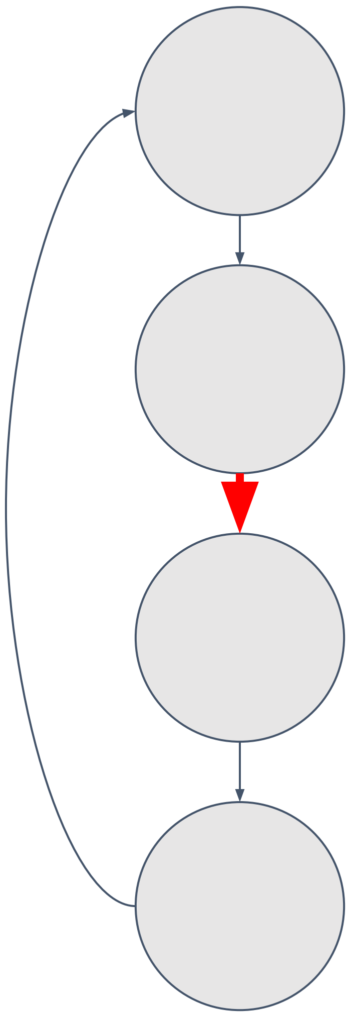
Availability

Match

Feedback

(Macvean et al, 2016)

Example problem: Discovery



Goal

I want to delete a file

Availability

Type “`File.`” and auto complete gives

```
void main() {
  File.
}         fromRawPath()
          fromUri()
```

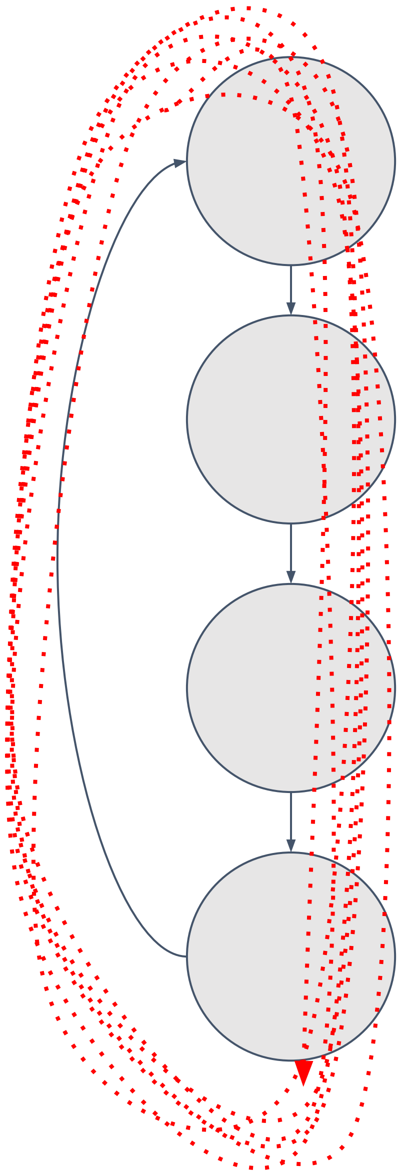
Match

There’s a conceptual mismatch on whether file is a static method or you have to get a file and then delete it

Feedback

(Macvean et al, 2016)

Example problem: 'yak shaving'



Goal

Availability

Match

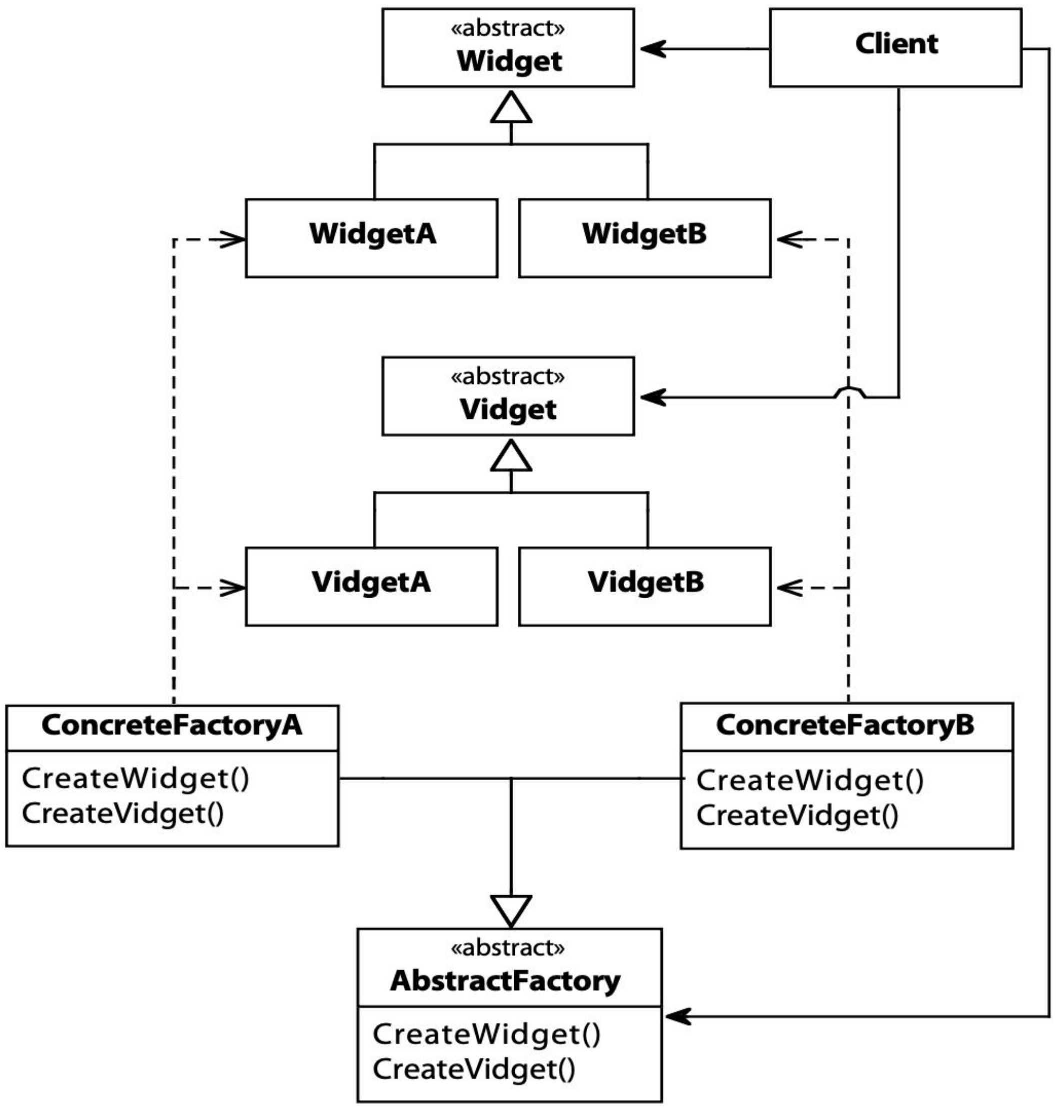
Feedback

To write a line to a file
Open a file
Complete a future to get the file
Convert a string to a bytebuffer
Iterate over the bytebuffer
Write the block
Complete on the future for writing
Close the file
Complete the future for closing the file

Too many subgoals that need completing

(Macvean et al, 2016)

Example (not-examinable)



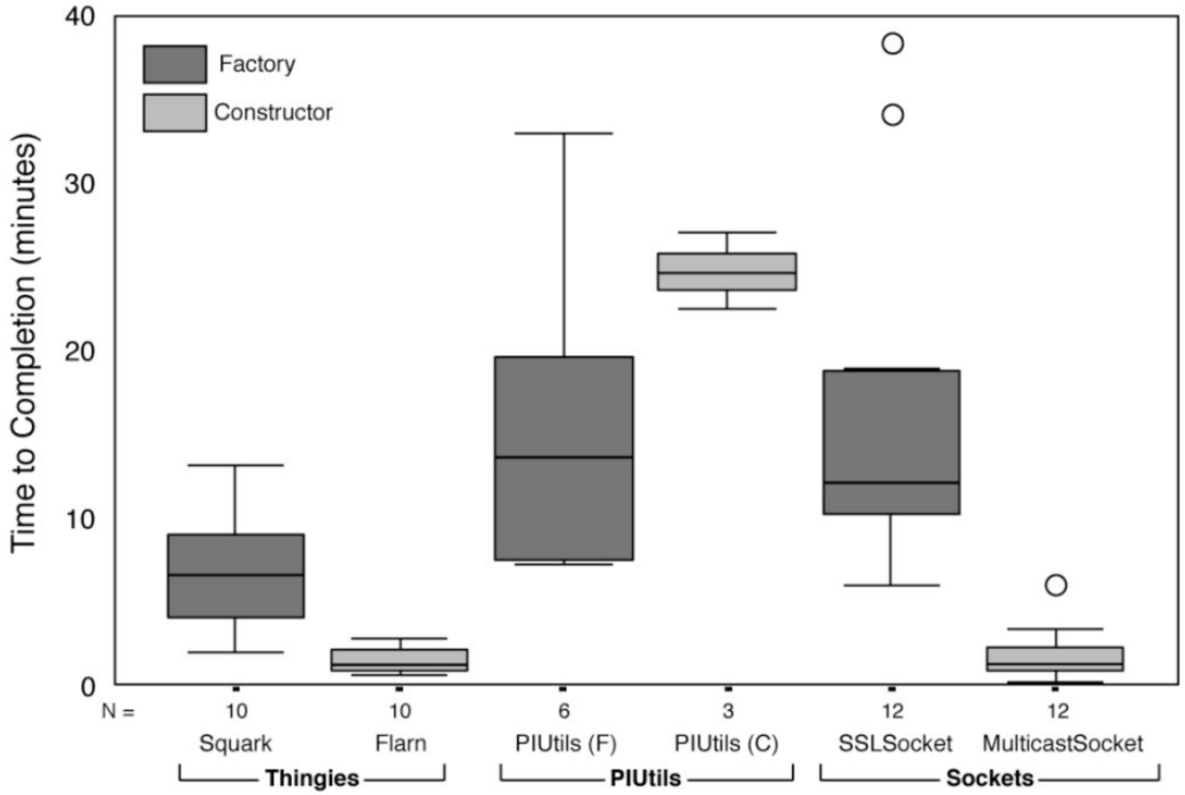
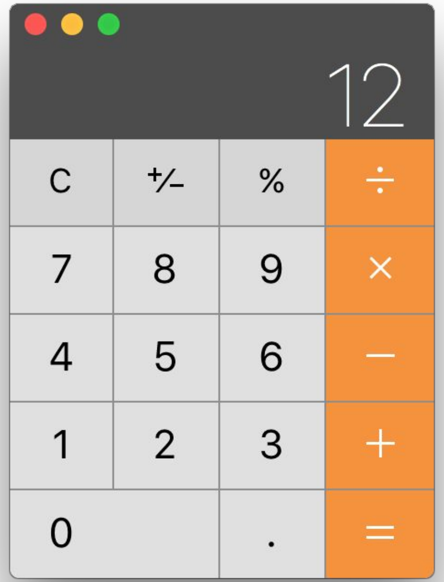
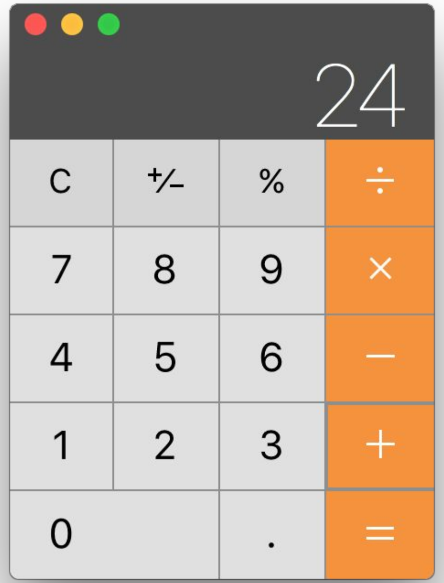


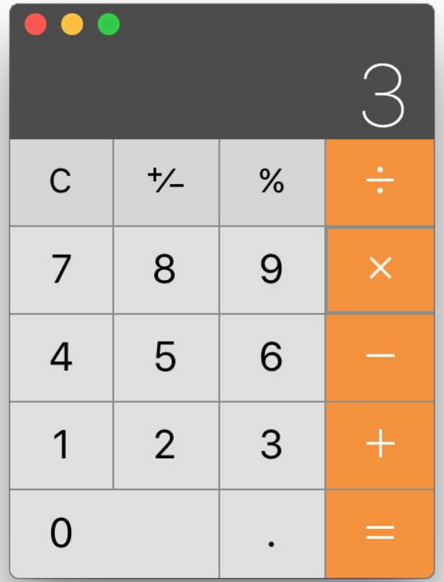
Figure 2. Time to Completion by Task

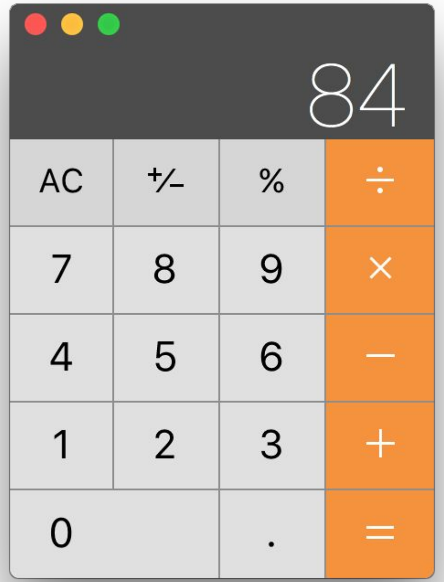
(The Factory Pattern in API Design: A Usability Evaluation, Ellis et al)

The cost of thinking:
Heuristics and Biases

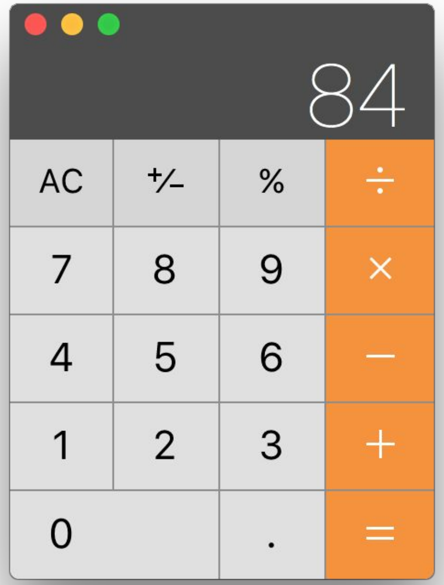

12
C +/– % ÷
7 8 9 ×
4 5 6 –
1 2 3 +
0 . =

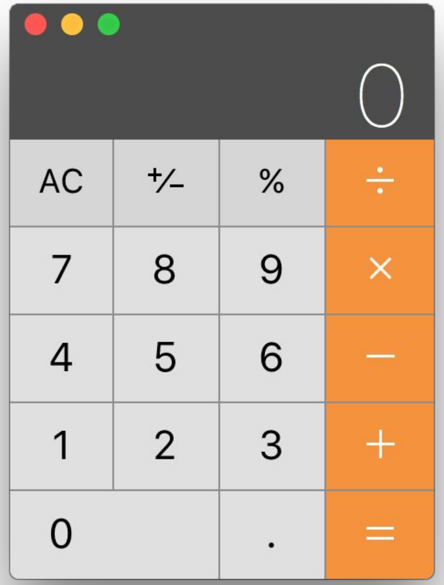

24
C +/– % ÷
7 8 9 ×
4 5 6 –
1 2 3 +
0 . =

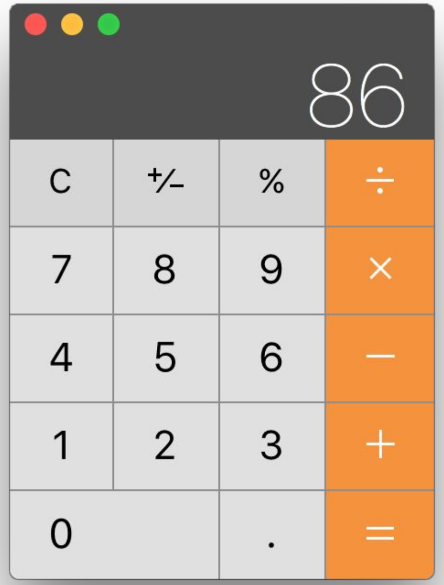

3
C +/– % ÷
7 8 9 ×
4 5 6 –
1 2 3 +
0 . =


84
AC +/– % ÷
7 8 9 ×
4 5 6 –
1 2 3 +
0 . =

12 + 24 * 3 =







= AC + 2

“eh?”

(Example from Richard Young)

How many times should the calculator user press AC?

Classical theories of metareasoning

- Optimal search
  - Find the best possible solution within stated constraints on resources
- Bounded rationality
  - Computation is one of the constraints
- Satisficing
  - Find a satisfactory solution within computation constraints

Neuro-economic models of reasoning

- Behavioural economics, sometimes applied as “nudge”

- Original basis in “prospect theory” (Kahneman & Tversky)
  - General theory of decision making
  - Construct a utility model, based on outcome of possible actions
  - Weight estimated values by likelihood
  - Choose action with optimal utility
  - May include future value discounting

- In practice, the optimisation is more likely to involve satisficing, due to reasoning with bounded rationality constraints
  - In Kahneman’s terms “thinking fast and slow”

Bounded rationality in humans

- Apply *heuristics* rather than searching for optimal plan
  - Availability heuristic - reason based on examples easily to hand
  - Affect heuristic – base decision on emotion rather than calculating cost / benefit
  - Representativeness heuristic - judge probability based on resemblance

- Apply *biases* to ensure estimation error within tolerable bounds
  - Loss aversion - losses hurt more than gains feel good
  - Expectation bias - researchers (even in HCI) find results they expected
  - Bandwagon effect - do what other people do

- And many others!

Behavioural economics in programming

- "Attention Investment theory" of abstraction use
  - Automation requires abstract specification
    - e.g. defining a regular expression for search and replace
  - Benefit of automation is saving time and concentration in future
  - But abstract specification (programming) takes time and concentration!
    - and more powerful abstractions (programs) can go wrong powerfully
  - User may *prefer* repetitive manual operations - safe and incremental

- So utility function will compare a) future saving of attention from programming vs b) costs of concentrating on a risky strategy
  - Biases such as loss aversion will apply
  - Bounded rationality will apply, since deciding what to do takes even more concentration

The limitations of goal based HCI

It assumes the user doesn’t make mistakes

- Would need a cognitive model of why error occurred
  - Information loss due to cognitive limitations
  - Incorrect mental model
  - Misleading design

- Need description of user journey that accounts for problem identification, diagnosis, debugging, testing, iteration etc

It assumes the user has the right goal

- Persuasive design is a field of HCI that addresses goal formation
- Applications:
    - Reduce energy consumption
    - Promote exercise
    - Manage diet and nutrition
    - Smoking cessation
- May include “nudge” to account for biases
    - But most people see this as paternalistic / patronising

It assumes the user knows what the goal is

- Not true when the purpose is a cultural/aesthetic experience (third wave HCI) - what is my “goal” in listening to a piece of music?
- Not true in “exploratory design”

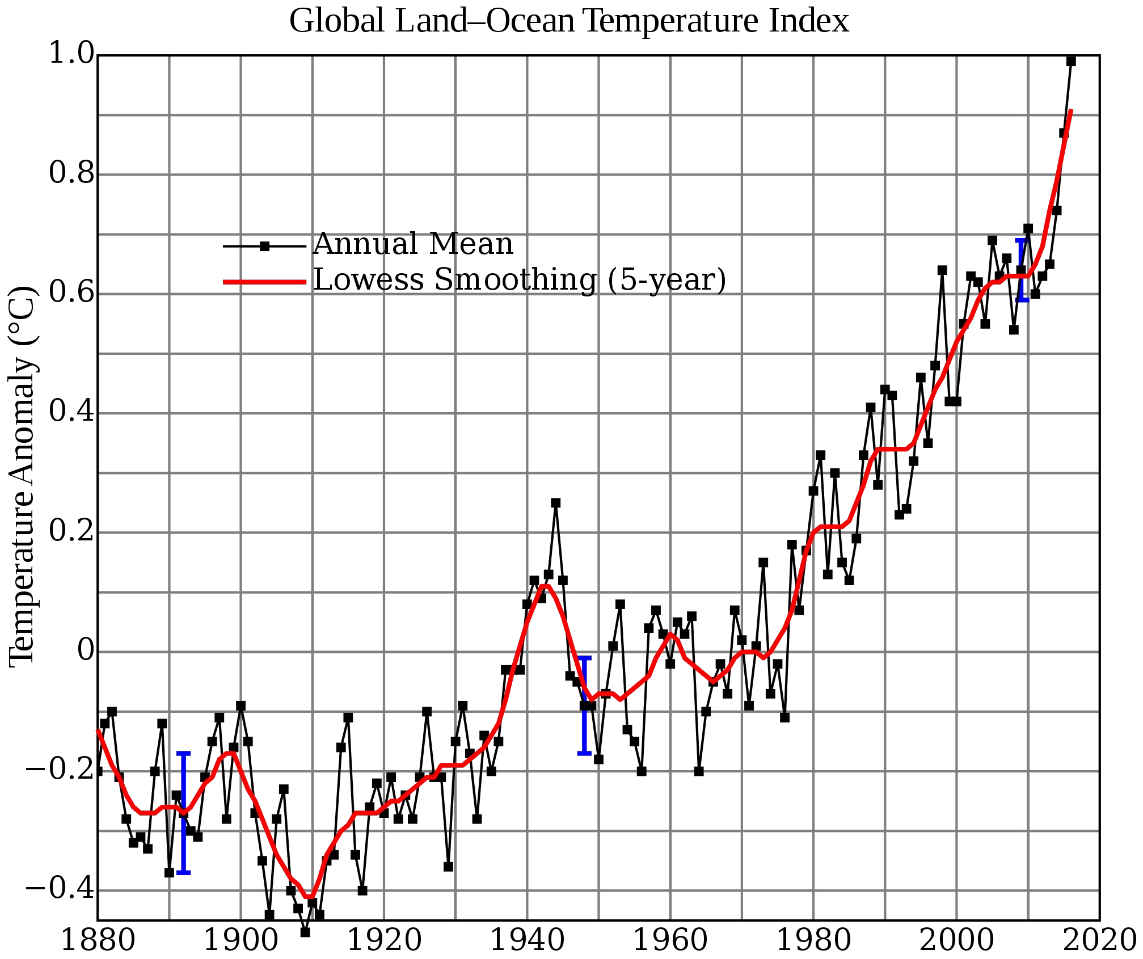
- More attention to this later in the course
- Some problems can’t be decomposed into actions
- Sometimes actions have side effects

Wicked problems

Including material provided by
Steven Tanimoto

A Wicked Problem:

Slowing climate change



By NASA Goddard Institute for Space Studies - http://data.giss.nasa.gov/gistemp/graphs/, Public Domain, https://commons.wikimedia.org/w/index.php?curid=24363898

More Wicked Problems

- Stopping the spread of antibiotic-resistant diseases
- Halting nuclear proliferation
- Ending homelessness in Cambridge
- Avoiding species extinction
- Colonizing Mars

Rittel-Webber Characteristics 1-5 of 10

1. There is no definitive formulation of a wicked problem
2. Wicked problems have no stopping rule
3. Solutions to wicked problems are not true-or-false, but good-or-bad
4. There is no immediate and no ultimate test of a solution to a wicked problem
5. Every solution to a wicked problem is a “one-shot operation”; because there is no opportunity to learn by trial-and-error, every attempt counts significantly

Rittel-Webber Characteristics 6-10 of 10

6. Wicked problems do not have an enumerable (or an exhaustively describable) set of potential solutions, nor is there a well-described set of permissible operations that may be incorporated into the plan
7. Every wicked problem is essentially unique
8. Every wicked problem can be considered to be a symptom of another problem
9. The existence of a discrepancy representing a wicked problem can be explained in numerous ways. The choice of explanation determines the nature of the problem's resolution
10. The planner has no right to be wrong

Challenge problem: Electronic Patient Records

Where healthcare meets data science ...

Option A) Give all your data to Palantir

Option B) Provide better end-user tools to doctors

- “A logical mind not a programming mind” (Blackwell & Morrison 2010)
- End-user tools for data-wrangling (Gorinova et al 2016)
- Uncertainty in bleeding management (Robinson et al 2022)