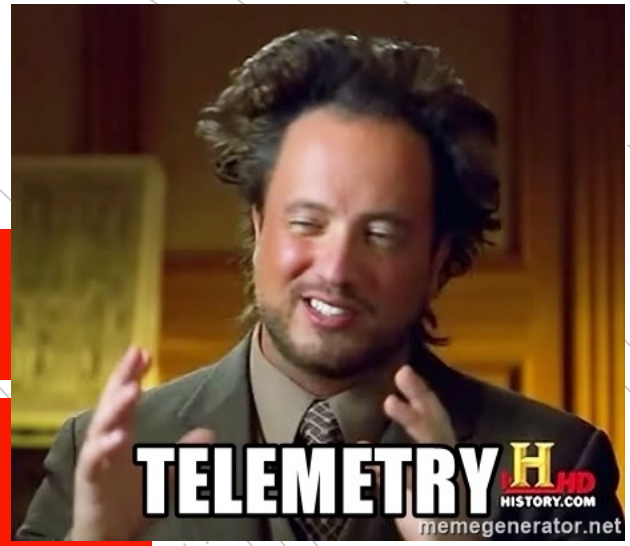
CSCE 438/838: Internet of Things

Internet of Things

- Enchanted Objects
- B2B Things
- Self-reporting IoT



Telemetry

Credits: Michael Hollman, hudl; Koopman, Better Embedded System Software

What is telemetry?

- "Telemetry is the science of gathering data [...] and transmitting this information to a distant receiver where it can be interpreted and also recorded for later analysis." - NASA

OK… but what is telemetry?

OK… but what is telemetry?

OK… but what is telemetry?

amazon

Google

OK… but what is telemetry?

- System
  - How your system/software is running
- Usage
  - What your users are doing
- Business
  - Who your users are

System Data

- Resource usage
- Request or operation rates/volume
- Error rates
- Crash reporting
- Specialized performance metrics
- Code execution timing/profiling
- Active database connections
- Infrastructural configuration
- Many, many more…

Usage Data

- Device rotated
- Internet connection established
- 43 seconds of inactivity
- Log in attempt
- Content accessed
- Upload succeeded
- Item added to cart
- Review submitted
- Skeleton spooked

User Data

1. Location
2. Age
3. Generation
4. Gender
5. Language
6. Education level
7. Field of study
8. School
9. Ethnic affinity
10. Income and net worth
11. Home ownership and type
12. Home value
13. Property size
14. Square footage of home
15. Year home was built
16. Household composition

Targeting options for Facebook advertisers - WaPo

User Data

17. Users who have an anniversary within 30 days
18. Users who are away from family or hometown
19. Users who are friends with someone who has an anniversary, is newly married or engaged, recently moved, or has an upcoming birthday
20. Users in long-distance relationships
21. Users in new relationships
22. Users who have new jobs
23. Users who are newly engaged
24. Users who are newly married
25. Users who have recently moved
26. Users who have birthdays soon
27. Parents
28. Expectant parents
29. Mothers, divided by "type" (soccer, trendy, etc.)
30. Users who are likely to engage in politics
31. Conservatives and liberals
32. Relationship status

Targeting options for Facebook advertisers - WaPo

User Data

33. Employer
34. Industry
35. Job title
36. Office type
37. Interests
38. Users who own motorcycles
39. Users who plan to buy a car (and what kind/brand of car, and how soon)
40. Users who bought auto parts or accessories recently
41. Users who are likely to need auto parts or services
42. Style and brand of car you drive
43. Year car was bought
44. Age of car
45. How much money user is likely to spend on next car
46. Where user is likely to buy next car
47. How many employees your company has
48. Users who own small businesses
49. Users who work in management or are executives

Targeting options for Facebook advertisers - WaPo

User Data

50. Users who have donated to charity (divided by type)
51. Operating system
52. Users who play canvas games
53. Users who own a gaming console
54. Users who have created a Facebook event
55. Users who have used Facebook Payments
56. Users who have spent more than average on Facebook Payments
57. Users who administer a Facebook page
58. Users who have recently uploaded photos to Facebook
59. Internet browser
60. Email service
61. Early/late adopters of technology
62. Expats (divided by what country they are from originally)
63. Users who belong to a credit union, national bank or regional bank
64. Users who investor (divided by investment type)
65. Number of credit lines

Targeting options for Facebook advertisers - WaPo

User Data

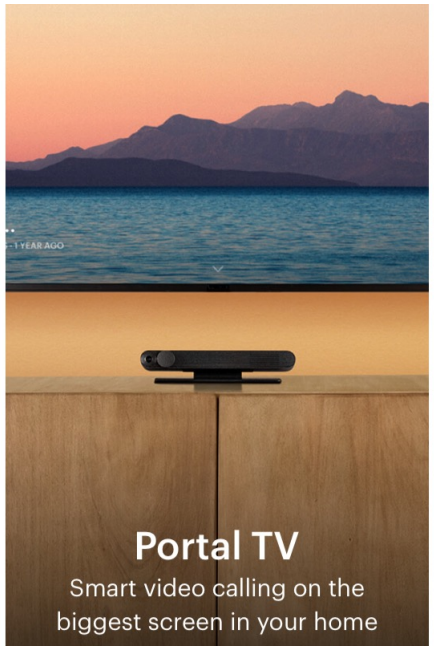
66. Users who are active credit card users
67. Credit card type
68. Users who have a debit card
69. Users who carry a balance on their credit card
70. Users who listen to the radio
71. Preference in TV shows
72. Users who use a mobile device (divided by what brand they use)
73. Internet connection type
74. Users who recently acquired a smartphone or tablet
75. Users who access the Internet through a smartphone or tablet
76. Users who use coupons
77. Types of clothing user's household buys
78. Time of year user's household shops most
79. Users who are "heavy" buyers of beer, wine or spirits
80. Users who buy groceries (and what kinds)
81. Users who buy beauty products
82. Users who buy allergy medications, cough/cold medications, pain relief products, and over-the-counter meds

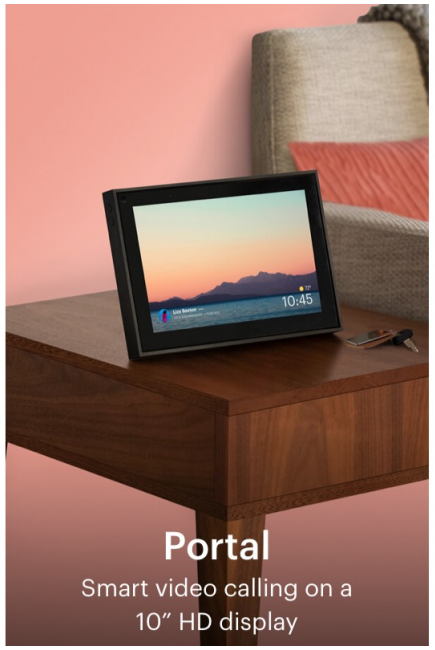
Targeting options for Facebook advertisers - WaPo

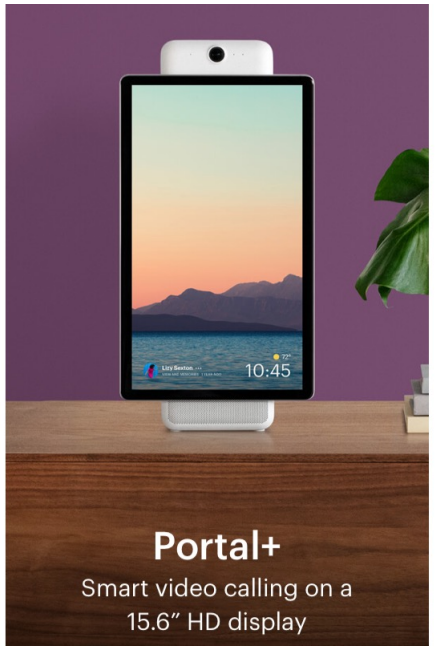
User Data

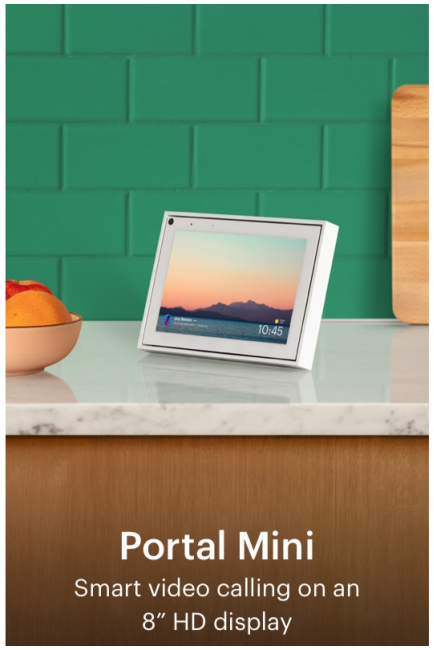
83. Users who spend money on household products
84. Users who spend money on products for kids or pets, and what kinds of pets
85. Users whose household makes more purchases than is average
86. Users who tend to shop online (or off)
87. Types of restaurants user eats at
88. Kinds of stores user shops at
89. Users who are "receptive" to offers from companies offering online auto insurance, higher education or mortgages, and prepaid debit cards/satellite TV
90. Length of time user has lived in house
91. Users who are likely to move soon
92. Users who are interested in the Olympics, fall football, cricket or Ramadan
93. Users who travel frequently, for work or pleasure
94. Users who commute to work
95. Types of vacations user tends to go on
96. Users who recently returned from a trip
97. Users who recently used a travel app
98. Users who participate in a timeshare

Targeting options for Facebook advertisers - WaPo





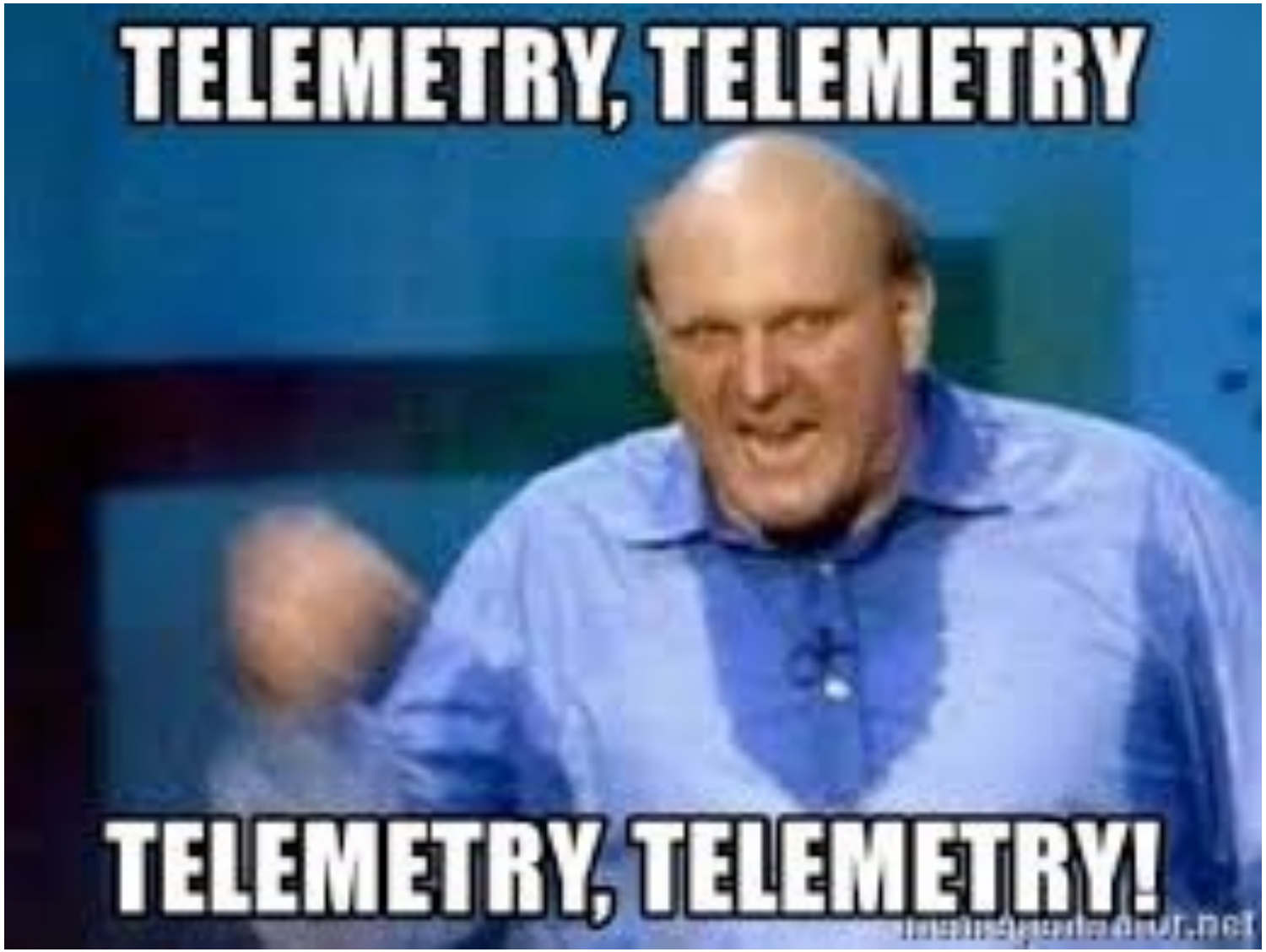




Facebook & IoT?

Blurring the Lines

- System
  - How your system/software is running
- Usage
  - What your users are doing
- Business
  - Who your users are
- No clear distinction
- Same telemetry data may be used for different purposes


TELEMETRY, TELEMETRY
TELEMETRY, TELEMETRY!

OK… but why telemetry?

- Clear, if it is your business model
- System data, usage data

Why telemetry?

- "The shoelaces won't tie!" - Your customer
  - Why?
  - Since when?
- "[Big customer] is threatening to cancel contract for [$$] because [your thing] doesn't work." - Your Boss
  - True?
  - Why?

Why Telemetry?

Telemetry helps you meet engineering needs and empower effective decisions

Engineering Needs

- Somebody shipped bad code
- The servers are on fire
- Hardware is not working and everyone is blaming your code
- We're being DDoS'd
- Error rates are through the roof

Product Decisions

- AB testing, experimentation
- Understanding usage patterns, user behaviors
- Identifying user groups

Strategic Decisions

- Historical data helps forecasting and goal setting
- Track more (and more meaningful) metrics

OK… but how?

Hard

Smart

Easy

Select a Difficulty

- Hard
  - Write your own stuff… all of it
- Smart
  - Wrap or integrate with external tools and services
- Easy
  - Plug & play tools and services

Select a Difficulty

- Hard
  - Write your own stuff… all of it
- Smart
  - Wrap or integrate with external tools and services
- Easy
  - Plug & play tools and services

Tools and Services

- Iotify.io
- Mosquitto
- Malaria
- Copper
- MQTTBox
- MQTT JMeter
- Gatling-MQTT-Protocol
- NeoLoad 6.8
- SmartBear LoadUI
- Micro Focus StormRunner
- Micro Focus LoadRunner
- Locust Paho Testing Utilities
- Check-MQTT
- MQTT-stresser
- MQTT-bot
- MQTT-PWN

Source: https://testguild.com/iot-testing-tools/

Logging

- Almost every platform alive has several great logging frameworks
- Set up a log aggregator
- Log with purpose
- Log consistently

Select a Difficulty

- Hard
  - Write your own stuff… all of it
- Smart
  - Wrap or integrate with external tools and services
- Easy
  - Plug & play tools and services

Telemetry for IoT

Telemetry for IoT

- Run-time error logging
- Watchdog
- System resets

Run-time error logging

Run-time Error Logging

- Need a way to find problems in (embedded) systems
- Things often fail in the field
- Directs the attention (blame) to the right component(s)
  - Software is not the only component that fails, but takes most of the blame
- A good source for problems that cannot be reproduced

How to Log

- Depends on the available resources
- Simple to complex
  - Reserve a single byte of non-volatile memory and store the most recently generated error code
  - Keep a queue of most recent error codes (8-16)
  - Extensive error logging
  - Streaming telemetry (Google Protocol Buffers)

What to Log

- Time stamps
  - Time stamp each entry (if possible)
  - Count since last reset
- System Resets
  - Source of reset
  - Frequency of resets
  - Uptime
- Run-time errors
  - Log error codes from run-time functions
  - Failure to allocate mem., stack overflow, communication port data errors, etc.
- Assertion violations
- Stack status
  - Log last few entries of the stack
  - Careful analysis leads to latest ISR or functions used before reset
- Hardware failures
  - Include run-time checks of hardware (I/O) for out of range values
- Non-computer equipment failures
  - Actuator cannot bring the system to a desirable state (detected through sensors)
- Operating conditions
  - Sensors data

What to Log

- Log everything that you think will help diagnose problems
- … and log some more
- If you know what problem you'll face, you may have already solved it

What to Log

- Time stamps
  - Time stamp each entry (if possible)
  - Count since last reset
- System Resets
  - Source of reset
  - Frequency of resets
  - Uptime
- Run-time errors
  - Log error codes from run-time functions
  - Failure to allocate mem., stack overflow, communication port data errors, etc.
- Assertion violations
- Stack status
  - Log last few entries of the stack
  - Careful analysis leads to latest ISR or functions used before reset
- Hardware failures
  - Include run-time checks of hardware (I/O) for out of range values
- Non-computer equipment failures
  - Actuator cannot bring the system to a desirable state (detected through sensors)
- Operating conditions
  - Sensors data

Logging Resets

- It is useful to record how long the system runs between resets
- Use ISR
  - Before reset, the system may jump to a high priority ISR
  - Record up-time within that ISR
  - May not work for all systems, may not work with loss of power (battery-powered, energy-harvesting systems)

Logging Resets

- Periodically log "up time" information
  - Update a non-volatile memory location (Flash) periodically (e.g., every hour) with system up time
  - After reboot (and system becomes stable), create an error log entry with the latest up time (e.g., with up to an hour uncertainty)

Run-time Error Logging Strategy

- Log when things work, infer from this information when things crash and reset

WATCHDOG

Typical Watchdog Operation

- When system is reset, watchdog is turned off
- Necessary startup functions are performed
- MCU kicks the watchdog for the first time
- Need to kick again before it counts down to zero
  - Make sure software cannot turn off or alter watchdog once it is started

NASA recommends using a watchdog and emphasizes that it must be able to detect death of all tasks

Watchdog

- Not a magic wand, but a very useful tool
- System is reset if a program takes unexpectedly long to execute
  - Expectations might be wrong
  - Faults that slow the system
  - Faults that results in a hang
- Too frequent ISRs
- Unintended infinite loop
- Corrupted data sources
- Loops that run longer than intended (memory)
- Hardware faults
- Does not help detect
  - Arithmetic errors
  - Conditional logic errors

Good Practices

- Make sure all tasks are executed between kicks
- Kick it in only one place
  - E.g., at the end of the main loop
  - RTOS: Make sure each task contributes to the kick
- Make sure a task cannot crash without tripping the watchdog

Good Practices

- Pick timer interval correctly
  - Too big: Room for system slow down
  - Too close to expected execution time: Occasional unnecessary resets
  - If you are not sure how long the program should take, then do not use a watchdog (or an embedded system!)
- Keep track of watchdog resets
  - Watchdog provides seamless error-prone operation
  - Try to find and log what causes the reset (LEDs, run-time error logs)

Not So Good Practices

- Do NOT kick the watchdog with a timer ISR
  - A hardware timer is used to kick the watchdog
  - The rest of your system might have died except the timer
- Do NOT turn the watchdog off after it has been turned on*
  - Turned off watchdog = No safety net
  - * Some exceptions apply

Heartbeat Timer

- Similar concept for distributed/networked systems
- Node sends messages once in a while to tell it is alive
- Since it is run through a timer ISR, may end up with similar problems
- Nevertheless, useful to rule out communication issues

SYSTEM resets

System Reset

- A tried and tested method to keep the (embedded) system sane
- Two issues
  - When and how to reset
  - Quickly get into safe and stable behavior after reset
- Two ways
  - Manual (external) reset
    - Hardware reset button, soft reset, power cycling
  - Automatic (internal) reset
    - Watchdog timer
    - When resources are exhausted – malloc error, assertions
    - Periodically for maintenance

Reset for maintenance

- Please reset your aircraft
- 400 planes
- Please reset your aircraft
- Reset by a meteor!
- Curiosity performs warm reset
- Misc

System Reset

- IMPORTANT!: Most of the time only the MCU is reset
  - Any peripherals controlled by the MCU may still be running during reset
  - MCU may not be aware of the state of the rest of the system after reset
  - Gather sufficient information after reset before asserting control

After Reset

- Make sure initialization process does not place the system into an unsafe state
  - The top of your code is not only executed with the system start but also (repeatedly) with the reset(s)
  - Design your code for resets (consider start-up as a special case, for when you know the initial conditions)
  - May require branched initialization
    - Different setup for the first power up and subsequent powerup cycles
    - Use a specific location in Flash (in information memory) to keep track of cycles

After Reset

- Design your code for resets
  - Start-up - special case, with known initial conditions

After Reset

- A good approach is: Do not assume much!
  - E.g., Set acceptable values before powering up output pins
  - Study the datasheet for initial values of registers (or if they are ever initialized upon reset)
  - Make sure enough information is collected before acting
- Sample all inputs, initialize moving averages
- It may have taken a long time between the last operation and reset (especially in battery-powered energy-harvesting systems)
  - "In an embedded system, a second may take two weeks"

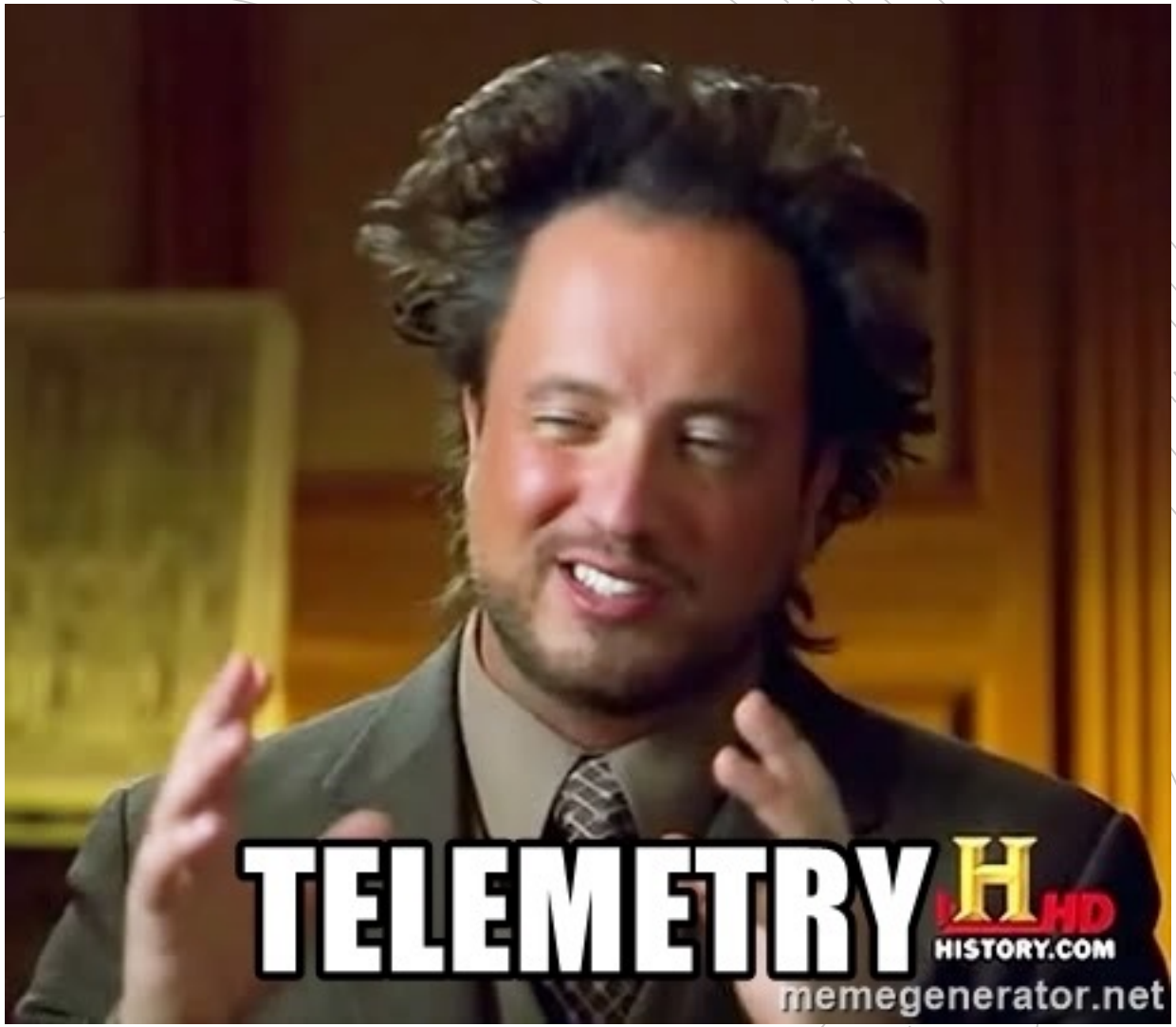
Two last things

#1

- You can make numbers tell any story
- Make sure you're telling the right one.

#2

- Respect user's data.
- Understand your legal limitations and ethical obligations.


TELEMETRY
HISTORY.COM
memegenerator.net