COVID-19 tracking apps, a year and a half later...

Stephen Farrell

stephen.farrell@cs.tcd.ie

November 2021

Background

- A March 2020 paper (*) asserted that: “A mobile phone App can make contact tracing and notification instantaneous upon case confirmation.”
- Most people would like that to be true
- My colleague Doug Leith and I were unsure if that was true or not, nor what privacy/security consequences might flow from population-scale uses of such Apps
- Trinity College Dublin (our employer) announced quick-turnaround funding for projects related to the pandemic (mostly aimed at medics)
- We applied for, and got funding for, Testing Apps for COVID-19 Tracing (TACT) Most of the clever stuff was done by Doug.
  - https://down.dsg.cs.tcd.ie/tact/

(*) Ferretti, Luca, et al. "Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing." Science 368.6491 (2020).

Bear in mind...

- I’m pretty critical of this overall system, but to be fair...
- In April/May 2020, most people felt that decisions had to be made extremely quickly and in the face of significant uncertainty
- We also believe everyone with whom we interacted on this topic was trying to do good
- But still: it’s important to improve/fix things that need fixing, even if that’s not always popular

Overview

- The Google/Apple Exposure Notification (GAEN) system
- A replay attack
- Measuring deployments
- Is Bluetooth proximity detection effective?
- What traffic do GAEN Apps send? (on Android)
- Conclusions

The GAEN System (1)

- GAEN == Google/Apple Exposure Notification
  - Details at: https://www.google.com/covid19/exposurenotifications/
  - There are other COVID-19 Tracking schemes, e.g. in Singaopre, Australia, but GAEN is most widely deployed
- A GAEN App uses a GAEN API implementation
  - App handles interaction between handset and Public Health Authority servers
  - API implementation handles “internals”: key storage, BLE beacons and beacon/key (TEK) matching
- Handsets generate a **Temporary Exposure Key (TEK)** every day
  - Those keys are ultimately intended to be uploaded to a public health authority's server if the app user has tested positive and agrees to upload
  - Other app users download the list of TEKs from the public health authority server(s) and can use those to detect if they were in proximity with someone who uploaded

The GAEN System (2)

- Bluetooth Low Energy (BLE) is a well deployed short-distance radio technology
- GAEN BLE beacons are sent @~4Hz and contain a Rolling Proximity Identifier (RPI) value that changes whenever BLE MAC address changes (about every 10 minutes)
  - $RPIK_i$:= HKDF(TEK_i,NULL,UTF8("EN-RPIK"),16)
  - $RPI_{i,j}$:= AES128($RPIK_i$,time-in-10-min-chunks)
  - Beacons also include encrypted TxPower value (unauthenticated encryption but not that bad here)
- Handsets listen for beacons for about 4s every 4 mins and record the beacon value and Received Signal Strength Indicator (RSSI) for 14 days
- If Alice tests positive, she causes the App to upload the set of TEKs she used for the last (up to) 14 days to her Public Health Authority server
- Bob’s handset downloads TEKs from his Public Health Authority server, perhaps every 2 hours, and checks if any TEK matches any stored RPI

The GAEN System (3)

- A number of interactions between Public Health Authorities have arisen over time...
- Within the EU, a number of countries share their sets of uploaded TEKs via a central server
- Within the US, a number of different states all use the same backend servers (so also share TEKs)
- TEKS from Northern Ireland are published by both Irish and UK servers
- In various places, “fake” TEKs were added to the real ones published in various ways, as an attempt at improved “privacy” (but a bogus attempt)

The GAEN System (4)

- Stated goal of these Apps is to detect if two handsets were within 2m for more than 15 minutes, based essentially on matching RPIs to TEKs and the associated RSSI and TxPower values
  - Spoiler: I don’t believe that can be reliably achieved
- “Attenuation” is the measurement used, defined as TxPower-RSSI
  - TxPower might be -27dB, RSSI might be -80dB so attenuation then would be 53dB
  - Measurements also need to be amortised over matching beacons seen and maybe (not specified) with some outliers thrown away
- GAEN API implementation accepts thresholds from App code and returns attenuationDuration values for the “above”, “between” and “below” ranges
  - E.g. App might input “[55,62]” and, if there’s an RPI/TEK match, get back “[10,3,11]” meaning handsets were “closer” than “55dB attenuation” for 10 minutes, in between 55db and 62dB for 3 minutes and “further away than” 62dB for 11 minutes
- App then decides if that output implies a notification is needed; If notified, user is usually guided to isolate, go get tested, etc.

The GAEN System (5)

- Governance: Google and Apple are in control
- (IMO) Reasonable justifications for that:
  - There are ~200 countries, don’t want wildly different 200 schemes
  - Google/Apple mobile OS duopoly, and their knowledge of handset internals, means you couldn’t credibly tackle any population-scale BLE scheme without them
  - Don’t want: fake tracing Apps, Apps draining battery messing with BLE or Apps using worse crypto than GAEN spec
- BUT: upshot is that OS updates can directly affect notifications without any visibility for Public Health Authorities (e.g. calibration adjustments)
  - That’s a governance issue: do we want Google deciding how to parameterise healthcare apps?

The GAEN System (6)

- Some interesting questions for these Apps might be:
  - How many people who would not have been found via manual contract tracing get notified?
  - How many people are notified sooner than would be the case with manual contact tracing?
  - How many notified people turn out to test positive for COVID-19, vs. the averages for testing at that time/in that locale?
  - What are the true/false positive/negative rates for notifications (where “true” == “within 2m for >15 mins” or whatever is the goal)
- So far, we’ve not seen the overall contact tracing systems (manual+App-based, together) being setup so as to be able to answer questions like those above. It’d be good if they had been.
- How many downloads or the proportion of the population who’ve installed isn’t anywhere near a good metric
  - “You need 60% of the population” is not true – that refers to an unrealistic model in Ferretti et al where the only contact tracing is via such Apps

Replay Attack

- There’s an obvious replay attack: collect beacons (likely from someone who’s positive) and re-tx/spread those to others elsewhere
  - https://down.dsg.cs.tcd.ie/tact/replay.pdf
- We calculated an (under)estimate for the amplification factor for such an attack
  - With conservative early-May 2020 Irish figures, “collector” at COVID testing station and “spreader” at hospital emergency department, each true-positive case could produce 4 or more false positive notifications
- Attack is obvious, hasn’t been mitigated and hasn’t (yet) happened (AFAIK)

Measuring Deployments

- The set of TEKs that Bob’s phone downloads are public, so we can download and count those (with a small bit of reverse engineering)
- Started doing that (hourly) in late June 2020 for Italian and German servers (Italy deployed GAEN 1st, Germany 2nd)
  - Each TEK is published for about 14 days, so our hourly downloads have many copies of each, but allows us to detect e.g. when things change
- We still download hourly, now from 33 services, including Irish, Swiss, Polish, Danish, Austrian, Latvian and various other servers...
  - https://down.dsg.cs.tcd.ie/tact/tek-counts/
  - I now (20211108) have 823 GB of TEKs
- Note: these measurements can’t tell us if these Apps “work” but might tell us they don’t work (in some places)

October 2020 Shortfall Estimates

- Analysis of downloaded TEKs indicated some significant shortfalls given **estimated** deployment densities and **estimated** number of TEKs uploaded, and numbers of cases actually declared
  - If an app is being used by 50% of the population in an area that declares 1000 cases on a given day, then we should expect about 500 TEKs to be uploaded
  - If instead we only see 400 uploaded, that’d be a 20% shortfall
- Details:
  - https://down.dsg.cs.tcd.ie/tact/survey10.pdf

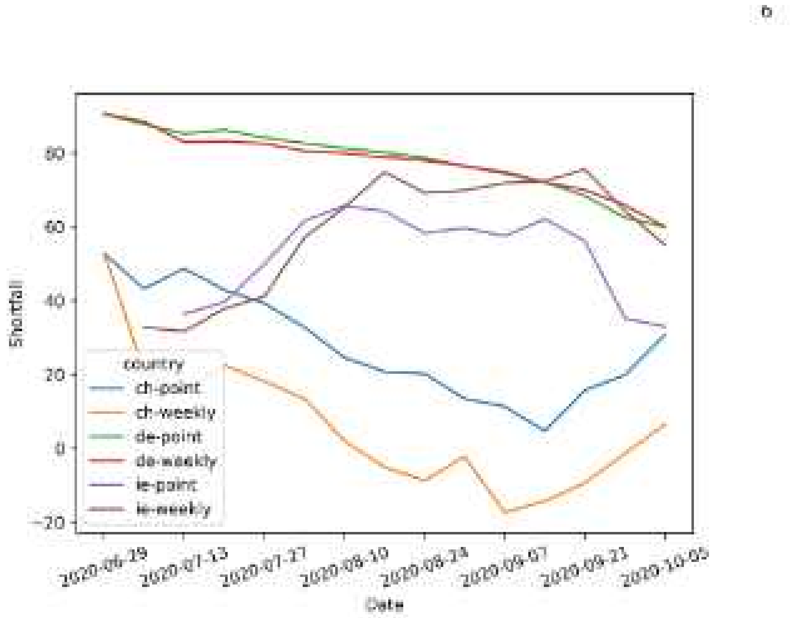
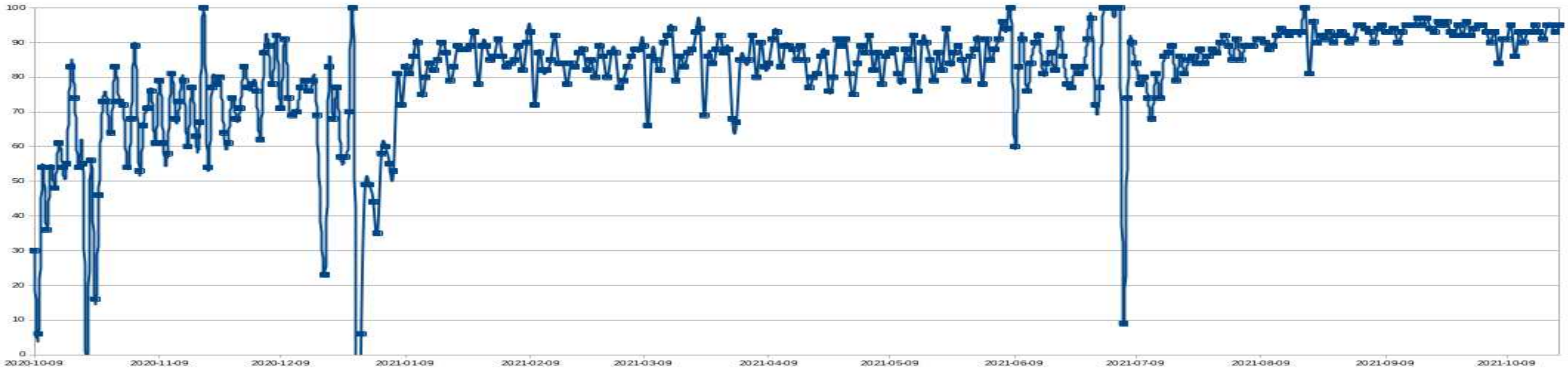


Fig. 4: Evolution of shortfall values as calculated using a point measurement of active user counts (“-point”) as used in Table I versus using install or active user counts (“-weekly”) from service providers as used in Table III.

October 2021 Irish Shortfalls

- As well as TEKs, Irish server also produces “statistics” for consumption by the Irish app – includes cases/county/day, but also includes the number of “active_users” and the actual number of new TEKs uploaded
- The “active_users” number has been precisely 1300000 since we started measuring these statistics (Oct 2020) but that is nevertheless the number of users that the HSE continue to claim are active (despite us asking in April 2021 and again recently)
- That leads to pretty bad shortfall numbers: 94% short for the most recent month
- Same calculation for April 2021 is written up at https://down.dsg.cs.tcd.ie/tact/ie-stats.pdf and actual data and scripts to reproduce are at https://github.com/sftcd/covidtracker-stats

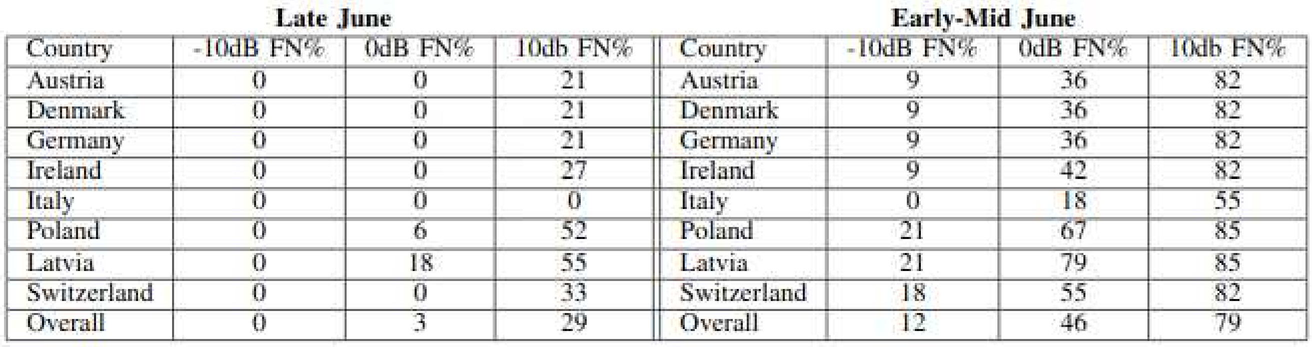


TEK Measurement Conclusions

- Even if all else had worked, there was unexpected friction when it came to getting uploads (systems integration isn’t trivial)
  - Significant shortfalls seen in multiple deployments for extended durations
- Not knowing how many active users exist makes it very hard to know whether or how effective these systems are (design flaw)
  - Knowing how many users exist needs to be done in a privacy friendly manner though, which is perhaps still a little beyond the state of the art

Does BLE Proximity Work?

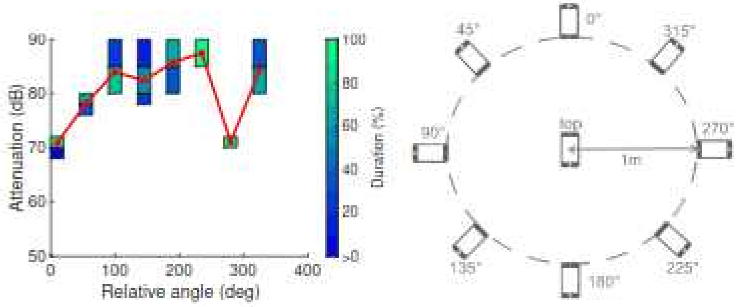
- 2020 pairwise tests with different handset types at 1m for 30+ minutes produced false negatives if you assume some “noise” due to orientation that affects attenuation
  - https://down.dsg.cs.tcd.ie/tact/posorient.pdf
- On June 13th 2020 Google shipped an update that added new calibration adjustments to attenuation calculations. We had tested before that, so we re-did the same tests. We haven’t re-done those since July 2020.
- Percentage false negative seen for various Country configurations:



Late June				Early-Mid June			
Country	-10dB FN%	0dB FN%	10db FN%	Country	-10dB FN%	0dB FN%	10db FN%
Austria	0	0	21	Austria	9	36	82
Denmark	0	0	21	Denmark	9	36	82
Germany	0	0	21	Germany	9	36	82
Ireland	0	0	27	Ireland	9	42	82
Italy	0	0	0	Italy	0	18	55
Poland	0	6	52	Poland	21	67	85
Latvia	0	18	55	Latvia	21	79	85
Switzerland	0	0	33	Switzerland	18	55	82
Overall	0	3	29	Overall	12	46	79

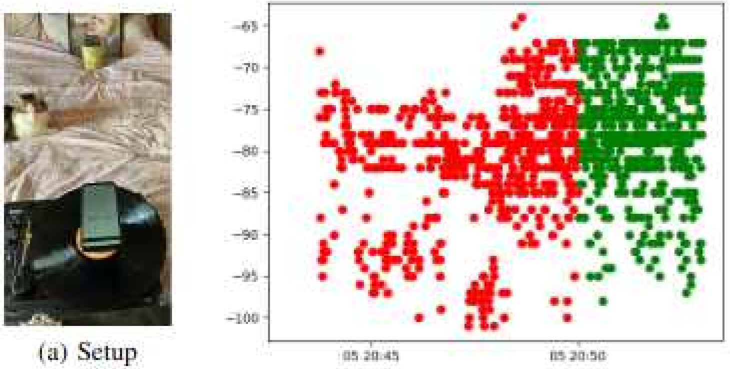
“Noise”

- Handsets package antennae in different ways and so orientation can change attenuation by itself (as can other things)
- We’re not sure how to model this but it seems to be able to affect RSSI (and hence attenuation) by 10-20dB



(a) (b)

Fig. 4: (a) Measured attenuation duration vs the angle between two handsets placed 1m apart. (b) Schematic showing orientation to which each angle corresponds.



(a) Setup

(b) RSSI

Cat video! https://down.dsg.cs.tcd.ie/tact/changes.mov

Real-World Scenario Testing

- We did tests of real-world scenarios to see how those affect RSSI, attenuation etc.
  - Walking, cycling, sitting around a table, on a park bench, between cars in parallel
- Two are noteworthy:
  - On a commuter bus https://www.scss.tcd.ie/Doug.Leith/pubs/bus.pdf
  - On a tram https://www.scss.tcd.ie/Doug.Leith/pubs/luas.pdf
- Those seem like scenarios where these Apps, if they worked, could help with contacts that would otherwise be missed but the metallic surrounds affected BLE distance 5 estimation badly

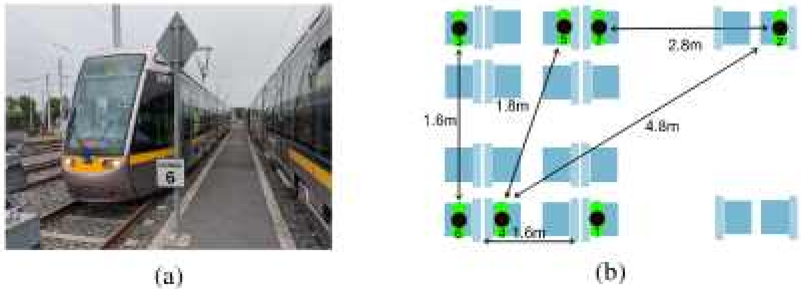


Fig. 1: (a) Tram on which measurements were collected. (b) Relative positions of participants during tests.

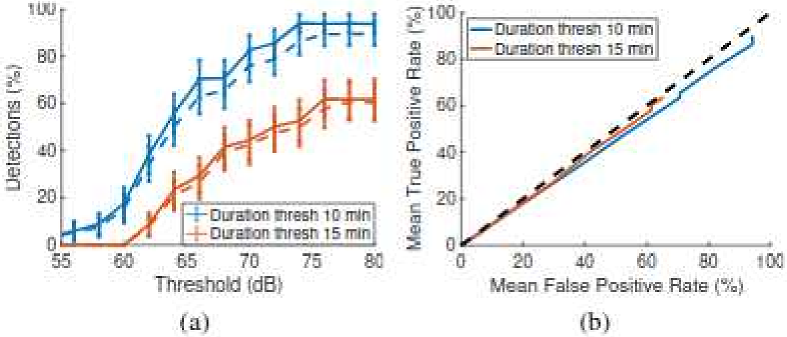


Fig. 8: Exposure notification true and false positive rates when a simple threshold strategy is applied to the GAEN tram dataset. (a) True and false positive rates vs attenuation level and duration thresholds, solid lines indicate true positive rates and dashed lines the corresponding false negative rates. (b) ROC plot corresponding to mean rates in (a), dashed line indicates 45° line.

What traffic is sent? (on Android)

- The Android implementation of the GAEN API is a part of Google Play Services. If you disable Google Play Services these Apps won’t work.
- In July 2020, we mitm’d and unpinned a set of GAEN Apps and captured traffic traces of App traffic and traffic from Google Play Services
  - https://www.scss.tcd.ie/Doug.Leith/pubs/contact_tracing_app_traffic.pdf
- Overall the Apps seem well-behaved but Google Play Services shares long term identifiers with Google and also frequently connects in a way that allows IP-based tracking

Example: Irish CovidTracker App

- For all apps, we only tested onboarding and TEK downloads – we didn’t touch anything that the client might e.g. do after a positive test as that could interfere with a running service
- Irish App had some issues:
  - An unnecessary “supercookie” – A JWT authToken used as a bearer token to download TEKs (no other services had such an individual cookie) – the HSE’s Data Protection Impact Assessment (DPIA) states that they don’t log that kind of information but it’d be better to not have it in the protocol
  - The App allows users to opt-in to sending “metrics” that mix devops (whether App opened that day) and medical information (how many times notified) – we recommend separating those into different security contexts
  - There was some path-not-taken code that makes a call to Google Firebase
- Most apps were open-source, had DPIAs, did certificate pinning and were pretty clean in terms of data-sent, Irish App would have been “mid-table” had this been a league in mid 2020

Google Play Services

- We turned off everything we could, including Google Play Services “Usage & Diagnostics” and tried to get to a minimal configuration where a GAEN App can run (reminder: that requires Google Play Services to be running)
- Roughly every 6 hours, Google Play Services connects to a “/checkin” API and sends back information including the handset IMEI, phone number, SIM serial number, WiFi MAC address and the email address associated with the handset
- Roughly every 20 minutes, Google Play Services connects to a “/p/log/batch” API including a cookie that is present in the “/checkin” HTTP request, thereby linking many long-term persistent hard-to-change identifiers with the IP address that can be geo-located
- The above messages contain additional telemetry – how much depends on settings and what other Apps are running (but is opaque); there was some variation in the frequency of connecting to the “/p/log/batch” endpoint, depending on settings.
- There is no published DPIA for the Android GAEN API implementation nor of Google Play Services (that we know about).
- Google Play Services is closed-source.
- That all seems hugely invasive to us and is required if you want to use a GAEN App on Android.

Doug Kept at it...

- “Android Mobile OS Snooping By Samsung,Xiaomi, Huawei and Realme Handsets”
  - In-depth analysis of the data sent by six variants of the Android OS, namely those developed by Samsung, Xiaomi, Huawei, Realme, LineageOS and /e/OS. (6th Oct 2021)
- “Comparing Privacy of Android and iOS: Mobile Handset Privacy: Measuring The Data iOS and Android Send to Apple And Google”
  - Systematic study of data that iPhones share with Apple and directly comparing data sharing by Apple iOS and Google Android. (25th March 2021, updated 10th June 2021)
- Those and more at: https://www.scss.tcd.ie/Doug.Leith/

Conclusions (1)

- Replay attacks (and perhaps others) exist and were not mitigated
- Significant shortfalls in uploads seen in multiple deployments
- BLE-based proximity detection may not work as well as claimed
  - Should improve over time, not sure if it will ever be reliable
- Overall design does not allow for privacy-friendly measurement of efficacy
  - That needed to be considered at the start, but was not

Conclusions (2)

- Android implementations of the GAEN API have serious privacy problems
  - If you don’t care about Google tracking you, then you have no problem; If you don’t want to install a GAEN App, then you have no new problem; If you care about Google tracking you and want to run a GAEN App, you have a new problem
  - There was some work to produce an open-source non-Google Play Services alternative GAEN API implementation, not sure if that was effective
- The mobile ecosystem exposes lots of sensitive user data. That really needs mitigating before encouraging population-scale deployments of healthcare apps like these
  - Population-scale is not the same as choosing to install candy-crush

Conclusions (3)

- The “<2m & >15 mins” criterion may be asking the wrong question – we find it hard to see that that can be reliably determined via BLE, we don’t know if some other (epidemiologically) useful criterion could be reliably determined
  - Perhaps aerosol infection turns out to be more important than droplets? (Droplet assumption seems to have having affected the 2m criterion)
- The rush to implement and deploy “within weeks” back in 2020 was misguided. Developing an app does not mean solving a problem and maybe some other app would have been better
  - One suggestion (from Serge Vaudenay I think) was to simply “randomly” ask the user to go get tested – with smart choices of random parameters, if that did work (who knows?) it would have been simpler and better in almost all respects
- Governments still seem to be encouraging adoption for entire populations, albeit much more quietly these days – maybe it’s time to stop that and explicitly tell people to turn all this off?

Thanks

Offline questions welcome too
stephen.farrell@cs.tcd.ie

These slides:
https://down.dsg.cs.tcd.ie/tact/tact-202111.pdf