



# building a connected car startup

- ▶ connected car, they say?
- ▶ what's there, already?
- the CAN bus
  - developed since 1983 by Bosch GmbH
  - initially invented to reduce the huge amount of cables to one cable/bus for all
  - a serial bus that connects *all* electronic devices in a car
  - message sending/receiving by priority
  - multi-master - all participants on the bus are equal

- ▶ connected car, they say?
- ▶ what's there, already?
- the OBD-II protocol
  - since the 1980s the California Air Resources Board (CARB) requires emission control capabilities for cars
  - it is a standardized I/O interface for accessing diagnostic data of control units of cars
  - but you can also access more vendor specific data, ...
  - the OBD-II diagnostic connector (SAE J1962) is required by law for cars built later than 2001 (gasoline) / 2004 (diesel)

- ▶ connected car, they say?
- ▶ what's there, already?
- and what is not there, already?
  - there is no concept of digital authentication or authorization
  - There is no login into a CAN bus
  - you simply need to have physical access in order to have access to the bus
  - if the doors of the car are closed, the access to the bus is closed

- ▶ connected car, they say?
  - ▶ ok, then let us do something with that
  - what do we have to play around with?
    - a built in SAE J1962 outlet in every car, required by law
    - the ELM327 protocol converter
      - a microcontroller translating 11 CAN-bus protocols ↔ AT-Commands
      - the code once leaked due to a not set copy protection flag by the manufacturer
    - bluetooth
    - smartphones
    - security by obscurity
- = high nerd value!

- ▶ connected car, they say?
- ▶ ok, then let us do something with that
- here is the plan:
  - we combine an ELM327 with a bluetooth chip set with a SAE J1962 jack and call that a „dongle“
  - oh, wait, we can already buy that hardware at low cost in large quantities
  - we will build apps that connect via bluetooth ↔ ELM327 ↔ OBD to all of the cars electronic control units
  - the play store / app store distributes our apps
  - we collect diagnostic data and thus know the condition that cars are *really* in and we offer remote maintenance
  - we will make this a platform and call it „the app store on top of OBD-II/CAN“

- ▶ connected car, they say?
- ▶ ok, then let us do something something with that
- the investor pitch is ready:
  - a great opportunity to connect most cars that are on the road already
  - app developers will develop their service apps for our platform, the platform gets its share
  - our next dongle (which we will build ourselves) has a far better protocol converter than the ELM327
  - the first mover will get to place *his* dongle in *each* car
  - the winner takes it all and hence *has* <sup>TM</sup> the platform <sup>TM</sup> of connected cars
  - big market, big bucks



- ▶ a great plan!
- ▶ but is it? really?



- ▶ a great plan!
- ▶ but is it? really? go talk to the business guys
- try to pitch the idea of an OBD-II company that builds a platform <sup>TM</sup>:
  - you don't own the vehicle that you want your user to plug you in. there is no vendor lock-in into *your* specific dongle.
  - you first must get into each car with your dongle. what is the initial value proposition for users to install such device in their car?
  - the insurance use cases? how about privacy?
  - you don't have any app developers to develop on your platform, yet?
  - you solve that app issue with money? you buy your reach, at least for the first 10x thousand users?

- ▶ still a great plan?
- ▶ and it is getting worse, when talking to tekkies
- what is also missing:
  - does that CAN bus really connect all control units in a car? no.
  - the CAN bus is no IP V4 network.
    - every read operation is also a write operation.
    - there are many protocols you can run on a CAN bus
  - security by obscurity is turned against you
    - You can read but what does the result mean?
    - vendors can and will build and change control units faster than you can reverse engineer them
    - there is just too many different CUs and their very specific access coding

- ▶ still a great plan?
- ▶ and it is getting even worse, when talking security experts
- the security nightmares:
  - you play out your firmware updates of the dongle via your smartphone ?!?
  - run any CAN bus application in a „read only“ mode is not feasible
  - the „what could possibly go wrong attitude“ is really bad at this particular point because
    - Interfering with 1500 kg of metal moving at 190 km/h (aka: a car) is clearly differentiating the impact from just hacking an e-commerce shop.
    - The most common and most successful attack vector is the firmware update, hence you should not compromise here.

- ▶ not a great plan!
- ▶ and now what? give up?

- ▶ not a great plan!
- ▶ and now what? give up? no.
- let go on that OBD-II/CAN thought and *shift your perception*:
  - look at the hardware/software progress in GPS-receivers
  - the system-on-a-chip approach can also be found in dongles
  - sophisticated dongles are in fact smartphones w/o screen and keyboard
  - and they are getting cheaper
  - and they are available with a SIM cards and as SAAS
  - and they come in a suitable form factor
  - do not waste time thinking of building your own hardware

- ▶ a new great plan
- ▶ the trip logging company
- the least invasive OBD approach:
  - only obtain RPM, current speed and VIN via OBD
  - also be ready to work entirely without OBD
  - create a trip logging that works on either one of
    - OBD-II data providers
    - GPS-only data providers
    - direct OEM integration endpoints data providers

- ▶ a new great plan
- ▶ the trip logging company
- do not call yourself an OBD company, instead, call yourself a trip logging company
  - concentrate on development of an excellent UI/UX
  - concentrate on one product, do that one right.
  - the „Fahrtenbuch“ became a success story, because
    - it is really plug&play
    - it is save to operate
    - it is fully compliant to the requirements of the tax authorities
    - it serves a *real* market need



- ▶ a solid advice that helped us, too - to grow to 45 employees in two years

„Rather than buy into the idea that you have to use your first decade to build your audience, collect your data, or lock in your customer THEN figure out how to make money, start by building a real business from the beginning.

The employees you don't have to fire, the users you don't have to burn and the early investor's ownership you don't dilute will thank you for it.“

<https://medium.com/strong-words/no-more-lost-decades-901027c6b1df>

- ▶ OK. check!
- ▶ we have successfully created a business :)
- one working product is enough
  - .. to make another one :)
  - based on a big user base of real users we now can
    - integrate more data providers of any source
    - build wide range of fleet management solutions&products
    - *credibly* be an „app“ in these future „app stores of connected cars“
  - based on our market standing we can now also focus on the direct OEM-integration

- ▶ some thoughts on what comes next
- ▶ identity management is key
- „connected car ecosystems“ require *really* good auth&auth
  - because the risk via OBD-II/CAN is high and the thread is real
  - fused key store on the embedded device to protect firmware uodates
  - preinstall the id management solutino ex factory, but that is up to the OEMs.
  - OEMs are to build and to operate these "app store for connected cars"
  - we build the apps
  - vimcar is ready whenever the OEMs are



# Questions and Answers

[lukas.weber@vimcar.com](mailto:lukas.weber@vimcar.com)





**vimcar is hiring**  
**[jobs@vimcar.com](mailto:jobs@vimcar.com)**