AUTONOMOUS DRIVING

TECHNOLOGY STACK

CONFIGURATIONS

IMPLICATIONS & INSIGHTS

5

LAYERS OF THE TECHNOLOGY STACK

MANU NAMBOODIRI & NITIN KUMAR

Autonomous Driving Technology Stack

Autonomous driving is coming of age and many companies are racing to find a place in the value chain. Each of them has a different approach implementing their technology stack - these approaches have profound strategic implications for costs, business models and use cases

AV Technology Stack: Layers and Components

At a high level, there are five primary layers of the autonomous driving technology stack i.e., sensors, infrastructure, compute and network, hardware/software interface and applications. Companies play within different layers of the stack to create value, and the resulting technologies come together as an ecosystem to enable autonomous driving. For example, Mobileye delivers vision-based sensor fusion, HERE creates maps and NVDIA enables compute etc. Apart from the core layers and components, there are critical technologies and policies enabling the autonomous driving value chain e.g., security, artificial intelligence, simulation technologies, various big data technologies as well as safety and regulations. While many of these are not explicitly called out inside the technology stack, they are implicitly embedded into every possible functionality of an autonomous vehicle (AV).

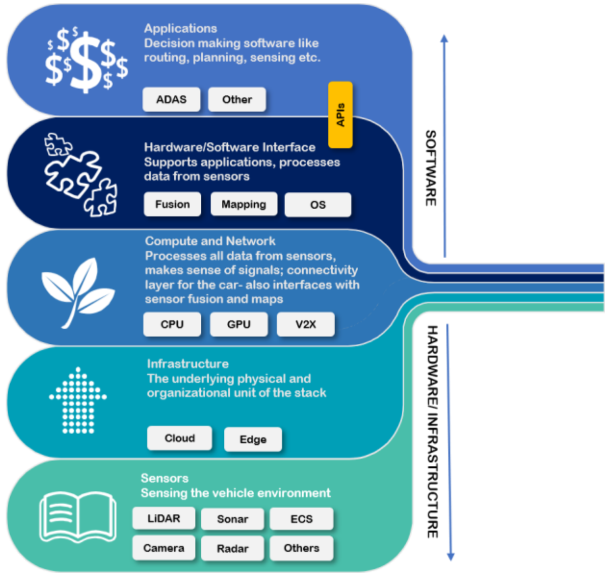

Figure 1: Autonomous Driving Technology Stack

These layers are responsible for sensing, connectivity, processing, analysis, and decision making, respectively. Sensors, infrastructure, and compute layers together form the hardware layer while network forms the connectivity layer, interfacing with sensor fusion and maps to pre-process certain types of data, enabling application to analyze and finally drive decisions. The APIs and interfaces for the software layers of the stack.

Let us explore the technology stack, its core components, their importance, and then look at implications of technology stack choices and gather some insights from it.

The AV Stack and its Implications

The technology stack architecture has widespread implications ranging from business models, use cases, vehicle efficiency, costs, safety security, go-to-market, etc. The following is a view of each element in the high-level AV stack and some implications based on choices made within each layer.

Sensors

The sensor package, within an autonomous vehicle is the foundational stack element to "see", sense and understand the surrounding context. A greater number of sensors capture more environmental data, increase component level redundancy but also add more hardware, fusion, interfaces, and compute costs. Fewer sensors may decrease costs and can potentially even boost driving range given less consumption of power attributed to sensor operation. For example, Tesla (cheaper cameras) while Waymo (expensive Lidar), is a topic of intense debate in the industry whether Tesla's camera only strategy will provide sufficient. Sensor type and numbers create or constrain degrees of freedom within your business models e.g., Tesla uses

cheaper vision cameras (higher data throughput) to enable Autopilot, other companies use Lidar technology adding costs and reliability issues. The upshot of this higher cost makes expensive AVs likely only suitable for fleet business models. However, the lower cost of sensors gives Tesla the option of selling the cars to consumers as personal vehicles, while keeping the option to also provide them to fleet providers like Uber and Lyft etc.

Sensor package design can also drive available use cases for AV providers e.g., a slower moving public transit bus running a fixed route with no highways, will not require long range external lidar, but requires the ability to process a lot more internal sensors (cabin cameras, wheelchair lock sensors etc.). On the other hand, a long haul AV truck must be capable of long range sensing, higher compute capacity and significantly shorter decision and response time, due to the higher freeway speeds. In addition, cargo versus passenger etc. use cases will also find their roots embedded into the choices and design of the sensor and compute stack.

Network

The network stack might be one that has lesser implications, since they are more standardized, having to interface with external infrastructures and standards. Cellular technologies have been gaining momentum in the US while DSRC based V2X has been deployed in Japan and Europe. Although both DSRC and C-V2X have been around and can talk seamlessly to 4G and 5G technologies, there are some inherent differences. The cellular technologies have advantages like communicating easily with pedestrians, cyclists who also have smartphones giving out precise locations. The C-V2X will be easier for retrofitting fleets or trucks without incurring high costs, DSRC will be more expensive to retrofit requiring global infrastructure improvements to enable V2I. C-V2X will also help reduce the cost of ancillary services like weather, congestion reports, traffic analysis etc.

Within the vehicle, there is a shift to IP based networking architectures like computers and data centers. Combined with the “softwareization” of the vehicle, a more modern internal network open up possibilities for better security, higher throughput, device virtualization and the ability to enable software updates and thus continuous improvement of the vehicle itself.

Compute

Requirements for compute are based on the amount of real time processing needed to sense the surroundings, analyze safe paths of travel, and

enable the timely execution of this plan. Compute will also be optimized for the type of workloads – such as video processing, AI decisions, and if multiple, including 3rd party, applications need to run simultaneously. Moreover, with the AV being the most complex edge compute environment, there is value in bringing proven cloud compute architectures to the edge – such as containers, data management, security models. All this impacts the design of cloud compute environment for AV.

Similarly compute configurations on the stack also have implications. For example, multiple HD cameras collect higher pixel data and potentially require more real time processing power, as compared to Lidar which collects lower quality depth images attributed to its wider environment sweeps. Tesla uses its own purpose-built hardware FSD chipset, focused specifically on its sensors and its own AI models, to optimize range while making cars smaller and cheaper. Waymo utilizes a heavier sensor stack and uses more general-purpose compute, potentially reducing efficiency and range of travel and therefore utilization. The tradeoff between sensor packaging has a second order impact on the level of compute needed as well as design factors.

Every decision related to the autonomous vehicle technology stack has profound implications on cost, use cases, vehicle efficiency and business models

Infrastructure

Compute and sensor choices create implications of the underlying infrastructure too. Edge compute choices have many implications e.g., ADAS versus full autonomous and personal versus fleet (higher data needs) etc. The edge architecture (both hardware and software) of the system constrains and dictates options available for providers. For example, Tesla’s edge compute architecture includes software updates, data capture, local AI inference, distributed search of interesting incidents the videos of which can be uploaded for training, among other needs. Lack of a capable edge architecture will limit monetizable use cases.

As automobiles progress towards full autonomy i.e., SAE Level 4/5, the distinction between ADAS (Level 3) and below become very clear. Many luxury cars such as Cadillac SuperCruise are purely ADAS, and do not have the edge computing needs like a fully autonomous system. A personally owned and operated vehicle will have much lower cloud compute requirements as compared to fleet operations.

Operating System

Operating systems are the heartbeat of any tech ecosystem and have been monetized differently by technology players. For example, Apple (iOS) charges a premium for hardware, Google (Android) drives more advertising revenue, while Microsoft (Windows) is monetized directly. Similarly, control of the OS will be vital to the autonomous driving value chain, as it impacts both customer experience and business models. Traditional automakers and software –first companies are in a race to create and control this OS. For example, while Tesla is mimicking the iOS model, Baidu, through its Apollo effort, is looking at "open source" AV technology to monetize ancillary products and services. On the other hand, Uber and Lyft are building proprietary AV platforms to enable their captive mobility use cases around rideshare and delivery.

The Tesla model will be aesthetically appealing to consumers given the hardware focus on exceptional design while the Baidu model will be hinging on utility while depending on demand for accessories, services, support etc. The right package and quantity of sensors will also depend on the OS strategy adopted, thus impacting value for ancillary players. Open and closed strategies for the OS e.g., iOS vs Android will also create consequences like safety legislation, competitive dynamics, standardization of infrastructure etc.

OS choices, including purpose-built such as Cruise or OEM platform independent like Aurora can also move organizations in one of many directions. For example, a company like Aurora is looking to be agnostic to vehicle platforms – with a goal to develop a general-purpose AV robot. The requirement to meet multiple use cases across trucking, rideshare, public transit can be daunting, but allows potential for a larger market share and return on scale. It is also likely that Cruise would be building to platform specifications provided by GM – and remain mainly focused on rideshare & delivery use cases to enable market scale for GM vehicles only.

Applications

While the rest of the stack drives the cost of the cars, applications or services running on top of the stack enable revenue components. One of the inflection points for applications will be the move from SAE level 3 to level 4 – i.e. from ADAS to AV applications. The competitive differentiation for both ADAS and autonomous applications, will be enabled by the underlying stack. Common ADAS features include auto steering, lane changing, auto braking, collision detection, adaptive cruise control etc., focused on assisting the driver, with user experience, driver feedback, stability and reliability provides differentiation. AV applications can also be differentiated by those enabling autonomous movement (i.e. fully replacing the driver) and those enabling mobility business services within the vehicle or aggregated at the fleet level.

Additional services can also be integrated at the application level, creating software defined packages for the cars, or leveraging APIs to build strong mobility ecosystems. One implication of the application stack for the business is the availability of Application Programming Interfaces (APIs). APIs are the ecosystem glue or independent monetization units for a company with implications on business models. A vertically integrated business and a partner focused ecosystem would build and expose APIs very differently, depending on whether applications are built exclusively for the manufacturer or if partner activity is encouraged on the platform. For example, partners can write apps which localize a service for a region or even provide a specialized service such as "rideshare for elderly passengers". When enabling partners, there can also be security implications to consider, specifically if third party applications can run and control non driving functions such as opening car doors, child locks, car windows, trunk etc.

> Security, simulation technology and artificial intelligence are embedded into every layer of the stack and play a critical role in every autonomous function

Critical stack enablers

Many areas cut across and enhance the stack either impacting internal functionality, an external value addition or constraint. For example, simulation is used widely to not only reduce the costs of live testing but also to validate against the long tail of incidents not possible to see in daily driving situations. Simulation is not a part of this stack but necessary to ensure the stack improves and provides consistent results in the real world. Similarly, AI frameworks are not part of the stack themselves, but are a critical component of how vehicles recognize patterns, images, predict pedestrian and vehicle behaviors etc. AI models resulting from training are an essential part of the decision flow in an AV system. Security is an intrinsic part of the entire stack –for example ensuring that the GPS inputs are not compromised, validating that video content is indeed coming from on-board cameras, ensuring secure connectivity and isolation of network and compute environments for partner applications etc.

Regulation and safety needs will also define and force choices on the stack design. For example, Nuro, the last mile delivery AV, was recently awarded the first exemption from federal safety standards i.e., not having rear view mirrors or windshields. Nuro can demonstrate that risk and safety are not negatively impacted, even without those features. Given, below 25mph speed in cities, suburbs, and no passengers - their sensor, collision detection, path planning software requirements are a lot lighter than passenger vehicles capable of driving at 65mph. The Nuro AV architecture will likely not be transplanted to any vehicle carrying passengers and safety and regulations will have implications on passenger versus last mile delivery capability.

Conclusion

While the high level architecture of the AV technology stack appears similar, the choices made within each stack layer has profound implications on costs, the target use cases, the business models enabled, the ecosystems or partnerships that can be cultivated, and regulatory/policy frameworks setting boundaries for optimal operations. The choices can be viewed as tea leaves predicting current and future strategies across the autonomous vehicle ecosystem.

About the Authors

Nitin Kumar is a 20-year veteran in the Hi-Tech industry. He is currently the CEO of Appnomic but played a variety of hands on executive roles ranging from CEO, Chief Growth Officer, Chief Transformation Officer, M&A Integration/Separation Leader, BU Head and Management Consulting Partner (corporate and PE portfolio companies). Nitin Kumar is a member of the Forbes Technology Council and shares his ideas and thoughts on the forum regularly. In his role as a former Management Consulting Partner, Nitin has done multiple strategy and M&A engagements for Software, Hardware, Semiconductor and AutoTech sectors gaining invaluable insights in the value chain, technologies, and business models. He is also a Certified Autonomous Driving Professional.

Manu Namboodiri has for 20 years worked across industries such as autonomous vehicles, security, IoT, software and has broad experience ranging from strategy, product, marketing, and ecosystem development. He resides in the San Francisco Bay area and advises companies in various stages of market adoption.