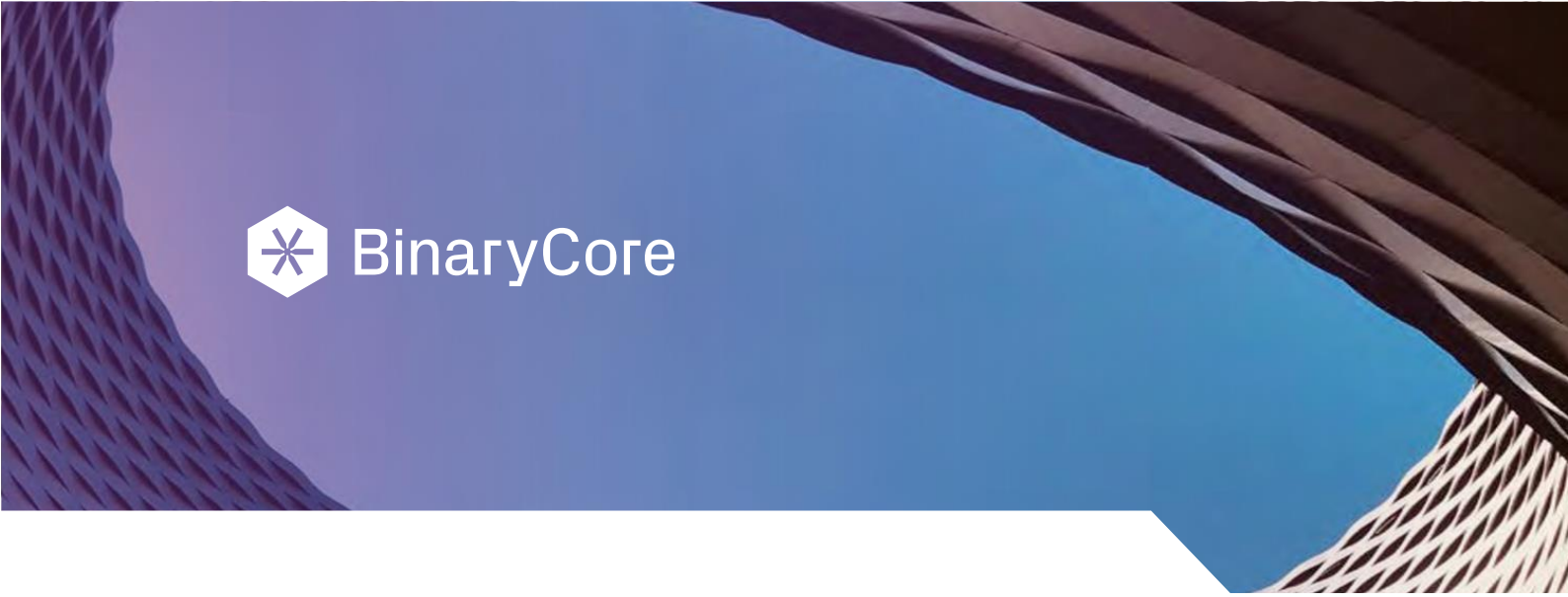
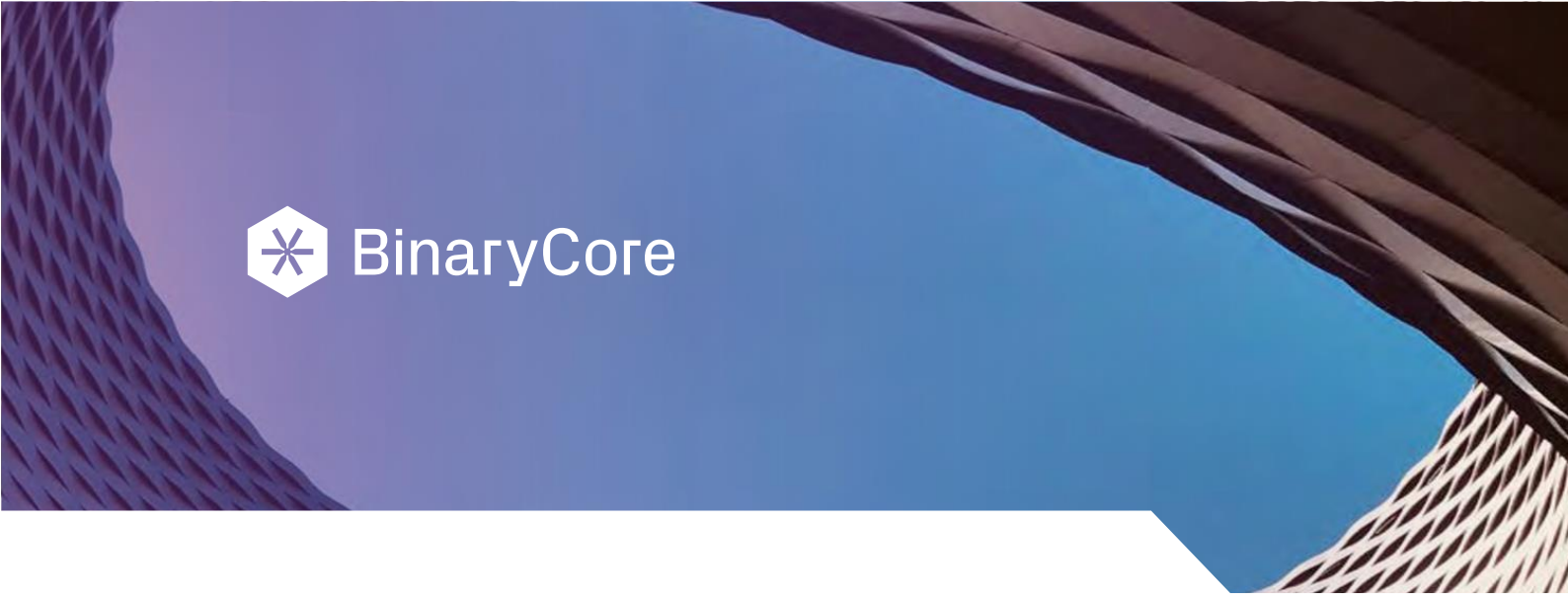


31 March 2022

How to calculate the software budget for your ideal product portfolio

To fulfil customer expectations and start closing the gap between traditional OEMs and software-native challengers, the strategic planners of today must define a future-proof software strategy and portfolio.

A key area of today's transformation in the automotive industry is the transition of original equipment manufacturers (OEMs) into software-driven companies. Leading OEMs are forced into this transition due to the crucial role software plays in meeting the latest customer expectations. In the eyes of most customers, vehicles with cutting-edge features retailing at competitive prices are now the standard. You do not have to look far to see examples of OEMs making major investments in this space. Volkswagen recently announced an investment of €89 billion into electric vehicles and software development over the next five years. Mercedes-Benz said it would invest €40

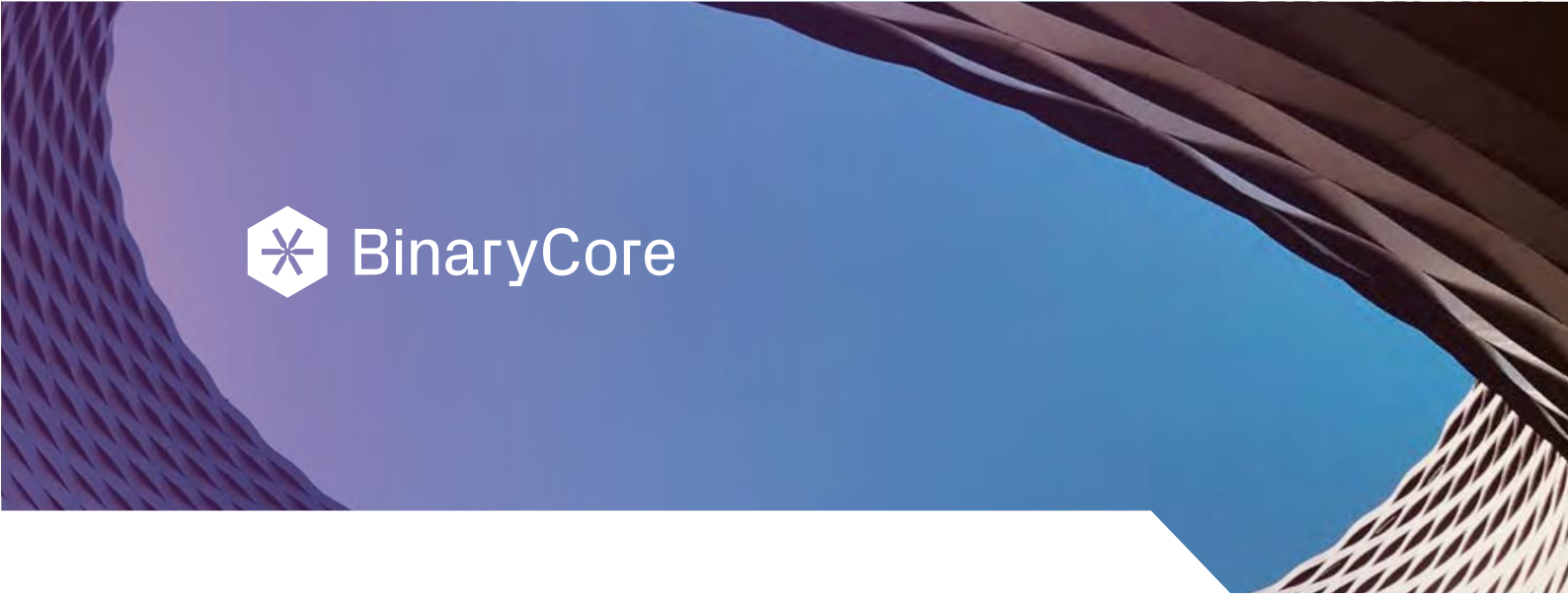


billion in electric cars, vans, and light commercial vehicles by 2030, and a further €5 billion into MB.OS.

Quantifying and prioritizing the right opportunities

While most OEMs are clear about the need for high software budgets enabling the continuous development and regular release of new features, they are less clear on what should be invested in and how to optimize the spend. For many, it's largely uncharted territory. The endless choice between different approaches as well as the often-limited software expertise in traditional automotive companies can make progress slow, or even send it in the wrong direction.

To fulfil customer expectations and start closing the gap between traditional OEMs and software-native challengers, the strategic planners of today must define a future-proof software strategy and portfolio. The approach must effectively quantify and prioritize the right opportunities, while also considering where resources are limited.



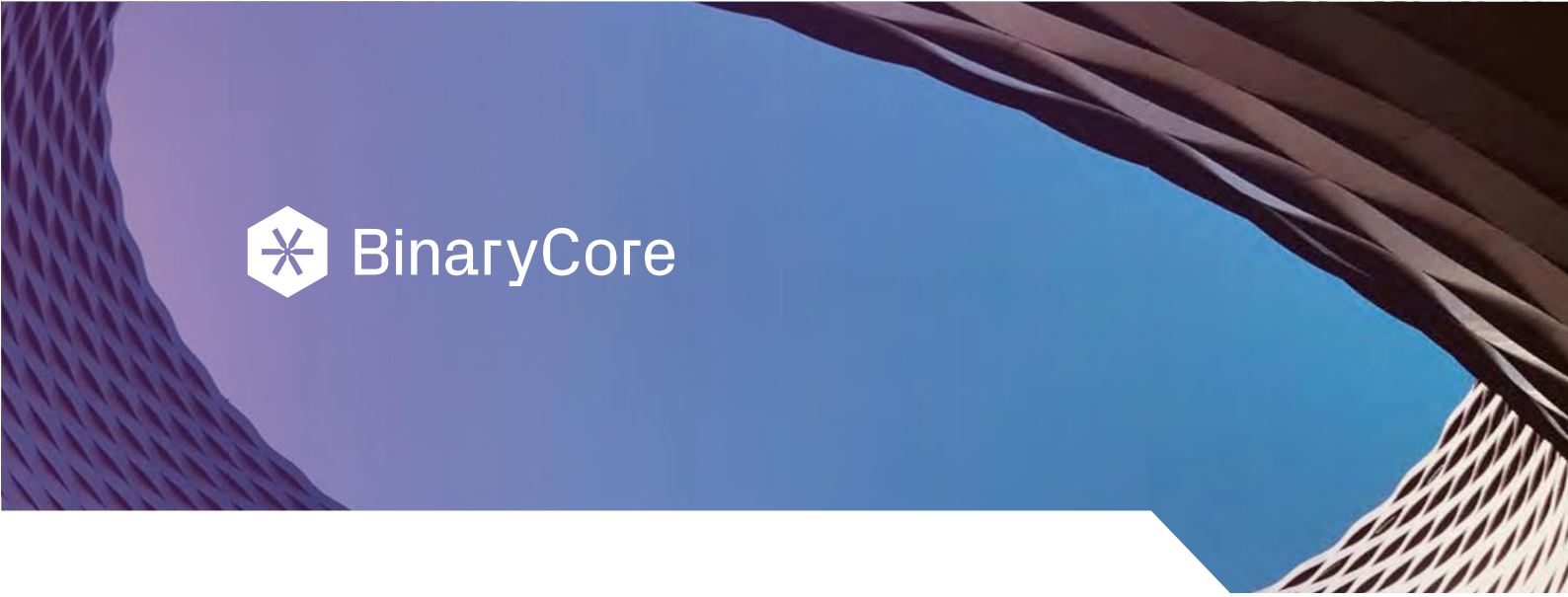
Kearney's comprehensive automotive software feature database

In response to more and more clients coming to us with this challenge, we worked with a range of suppliers and external experts to find an answer. We wanted to provide a concrete way of supporting our clients through a qualitative and quantitative assessment and prioritization of the software features in their product portfolios.

We developed a comprehensive database compiling more than 900 software features (everything from torque management to payment integration), covering in-vehicle and back-end domains on both an application and base operating system (OS) level. By adding different filters, the database compares features' characteristics, cost estimates, and potential optimization levers to identify the most efficient and effective solution. Our clients can now create an initial bill of materials and business case for feature development with moderate effort.

The database can be used in the following ways:

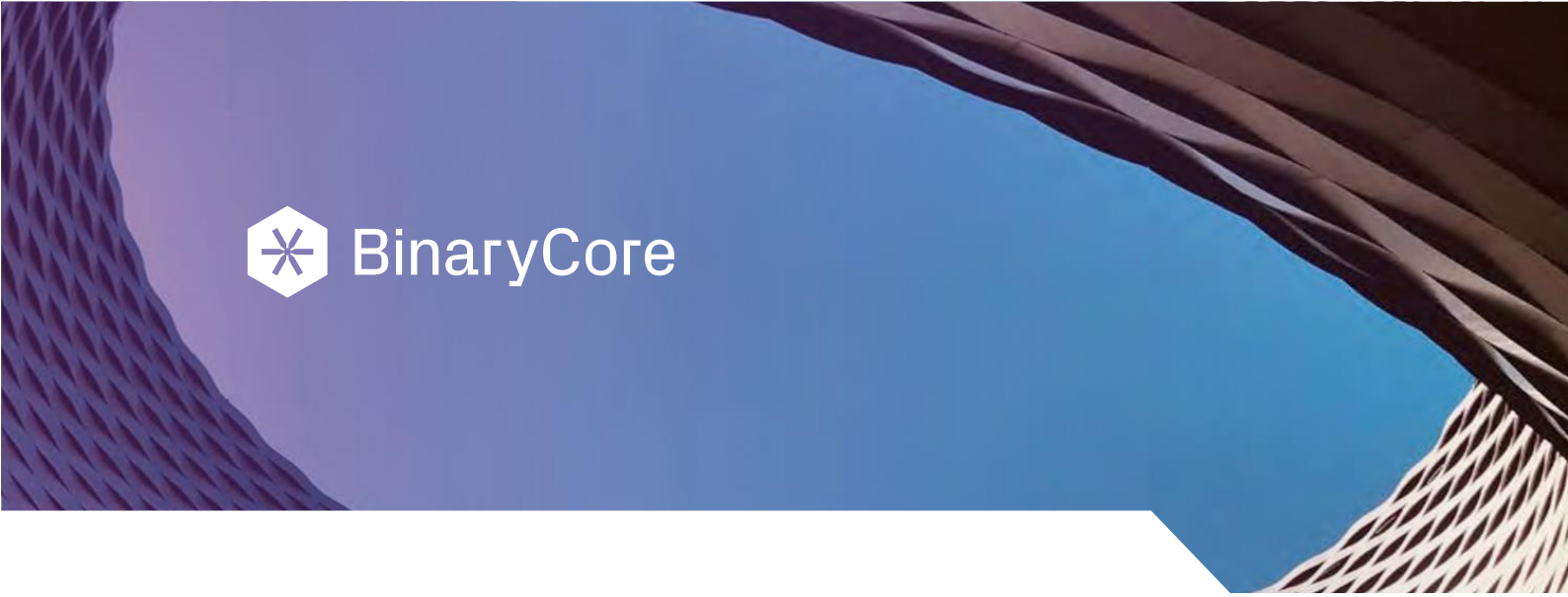
White spot identification. It's quick and easy to compare your current software feature portfolio with the database and uncover key features you may be missing in your software plan.



Feature analysis. Each feature included in the database is characterized by a set of criteria assessing the difficulty of implementation. These criteria are:

1. **Life cycle.** There are three defined life cycle stages: common, differentiation, and innovation. These denote the different impacts each feature will have on the value-add for customers.
2. **Technology stack.** There are three ways of assigning whether an organization already has the expertise required for new features or whether it will need to be leveraged from other industries: common, automotive, and high tech.
3. **Implementation confidence.** There is a scale from low to high rating the quality and confidence of automotive and tech players in feature implementation.
4. **Automatic Automotive Safety Integrity Level (ASIL).** The ASIL considers functional safety implementation and testing efforts.

Cost estimate and validation methodology. For budget planning and prioritization, each feature must have a clear cost estimate. We provide an initial estimate based on expert interviews and our best fit should costing methodology (should costing is an analytical method that benchmarks expertise to compare developer and practitioner estimates for features that have not yet been developed).



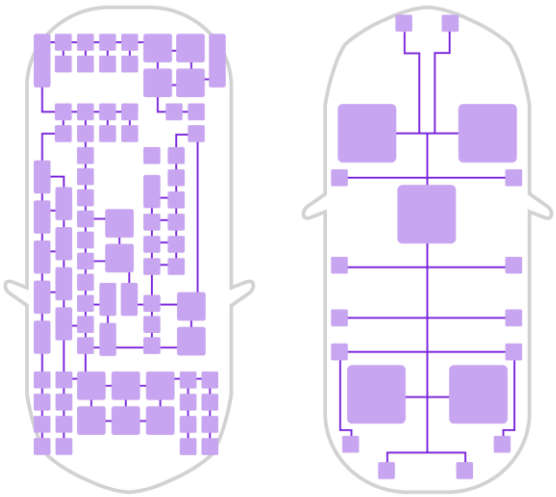
Strategic make-or-buy decisions. Once you've assessed the features' life cycle, costs, and customer value-add, you're in a good position to make strategic make-or-buy decisions. The choice of sourcing partner (for example, tier 1 suppliers or tech players) will have a significant impact on revenue shares in a buy case.

Optimization levers. Optimization levers can be used as hypotheses to further improve feature planning and strategy. As examples, these might include adjusting shoring models to deliver commodity features offshore, or the reprioritization of features based on revenue estimates.

How to use the software feature database for a platform cost comparison between two architectures

Due to decentral or domain-based E/E architectures, some OEMs are easily running on 60 to 120 ECUs, sensors, and actuators in one car. Below we outline how the database helps quantify the cost and savings of moving to a zonal architecture with fewer ECUs and a shared operating system (see figure 2).

Figure 2
Example of how the database helps quantify the cost and savings of moving to a zonal architecture with fewer ECUs and a shared operating system

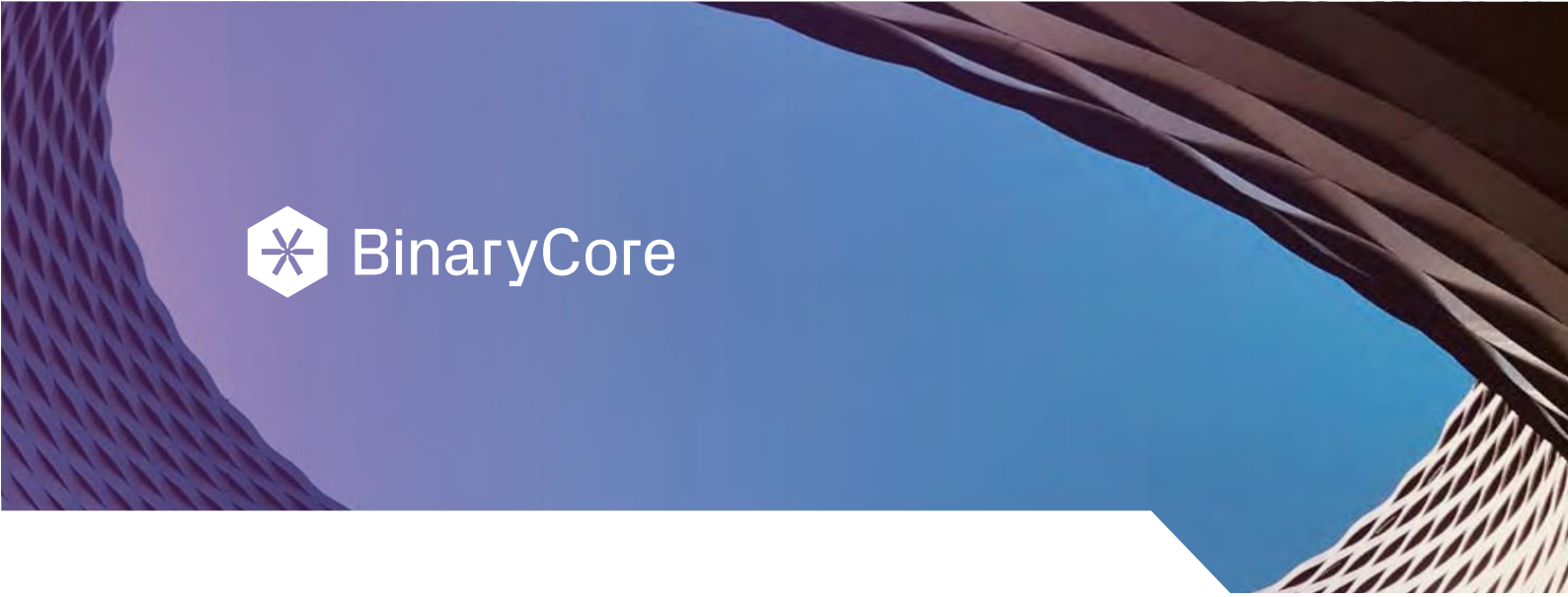


Source: Kearney analysis

An OEM wants to compare the base OS costs for a domain-based architecture versus a zonal architecture. They can set a filter in the database to electric vehicle, in-vehicle domains, and exclude sensors/actuators, leading to around 40 electronic control units (ECUs) at €115 million base OS costs for a domain-based architecture.

For the zonal architecture an additional filter can be applied on high-performance compute module readiness, which then reduces the required ECUs to five high-performance ECUs at either €5 million costs for one common OS or €25 million in the case of individual versions.

This clearly illustrates the potential and serves as an initial indicator in the business case of a zonal architecture for platform cost reductions. In a second step the cost increase in



software management and project management office can be quantified to sharpen the calculation. In general, the reduction in platform costs opens budget for further improvements at the application layer, ultimately enhancing the customer experience.

To discuss access to our software features database please contact socialmedia@binarycore.com

▸ Authors

Felix Kreichgauer, Marcin Kurzal, Michael Roemer, Sebastian Werner