TRANSFORMA INSIGHTS

revenera

Driving recurring revenue with software

Driving recurring revenue with software

Technical equipment used by enterprises in many verticals is becoming increasingly sophisticated. In decades past, industrial, medical, and many other types of machinery and assets would often be highly sophisticated from a mechanical perspective, but significantly less sophisticated from a software perspective. On-board software may have only related to the basic functioning of a machine and reporting current status to a control panel. It would have been infrequently updated, if ever, since the associated machinery was generally inflexible.

In today's markets, however, software capabilities are often a key aspect of an industrial asset, with such assets benefitting from a CI/CD (continuous integration/continuous delivery) approach to on-board software development, continually upgrading and enhancing the capabilities of an asset. This dynamic is particularly relevant in the case of Artificial Intelligence (AI), where any AI capabilities that may be present on-board industrial assets can often benefit from frequent upgrades and refinements to reflect enhanced learnings generated by cloud analytics engines that ingest data streams generated by multiple similar assets.

In addition, many industrial assets are now IoT-enabled and so can communicate data to remote locations for analysis, and also potentially accept control information from remote locations. Accordingly, IoT can enable the updating of on-board software from remote locations and also potentially enable new charging models and value propositions for end-users. This report discusses ways in which software on-board industrial assets can be managed, the benefits associated with remote software administration, and some of the key enabling technologies that are required to support new emerging software-centric propositions.

The benefits of remote software administration

Remote software administration for industrial assets can be used to ensure that on-board software is always up to date, that any appropriate software patches or software upgrades have been applied correctly, and that licencing conditions are appropriately enforced. This is a deceptively simple concept but unlocks many benefits, as discussed in the following paragraphs.

A significant benefit relates to the enhanced value propositions that remote software updating enables. Each individual machine, or asset, present on a customer's premises can be individually configured in terms of the software capabilities that are enabled, and the availability of such capabilities can be managed over time or on the basis of usage, or depending on a range of other parameters. Accordingly, software management allows for flexibility in charging models, including pay-per-use, or the application of sophisticated licencing frameworks across multiple assets. It also allows for 'try before you buy' type arrangements, or the provision of a specific set of capabilities for different users on a client site.

This kind of approach to software monetisation can be deployed to enable the servitisation of the associated asset, so that it is charged for on a 'service' basis according to usage, rather than necessarily purchased outright by the end-user. Reconfiguring payments for the provision of an asset from a one-off purchase to an ongoing stream has significant cashflow benefits for any end-user and significantly de-risks the adoption of new servitised solutions when compared to assets purchased in a more traditional way. End-users may also find that it is easier to upgrade (or downgrade) servitised solutions as their needs evolve, since a capacity upgrade can be enabled by a contractual change rather than implying a need to sell existing equipment and then purchase new higher-capacity equipment.

Another key benefit is that software updates (or upgrades, or patches) can be delivered quickly, from a remote location and without any need to access the equipment in question. This is particularly beneficial in scenarios where industrial equipment is located on the premises of an end-user. In such cases, and without a remote management capability, an equivalent software updating process could involve a site visit by a field engineer which, in turn, would require site access to be negotiated and agreed with site administrators. Alternatively, software updates could be distributed by shipping pre-loaded CDs, or via file transfer, although both of these approaches rely to some extent on client staff at a client site to implement the software upgrades, which introduces a significant potential risk of process failure.

Cost savings can be particularly significant in cases where remote software management replaces an existing process that relies on visits by field force engineers to update software, since field engineers will no longer be required to visit individual assets to upgrade the on-board software. This is particularly impactful in situations where assets are thinly distributed over wide areas (for instance, MRI scanners in hospitals) and accordingly travel distances associated with field visits are long. Corresponding sustainability benefits are also significant, due to the reduction in vehicle miles travelled.

In addition, remote management of software tends to be more robust and less error-prone than equivalent processes that rely on site visits by field engineers. Central systems are better able to maintain an accurate record of the software installed on any particular device, and also previous software configurations, to establish a 'digital thread' describing the software history of any particular asset. This can be a particularly valuable capability in the case of medical equipment where software configuration information may need to be communicated to a regulatory authority. Associated with the concept of a digital thread is the potential to roll-back software versions, should this ever be needed.

Remotely managed software upgrades can also be planned and implemented more quickly, so that software upgrades can be implemented at times when the corresponding asset is not needed, or during a period of scheduled downtime.

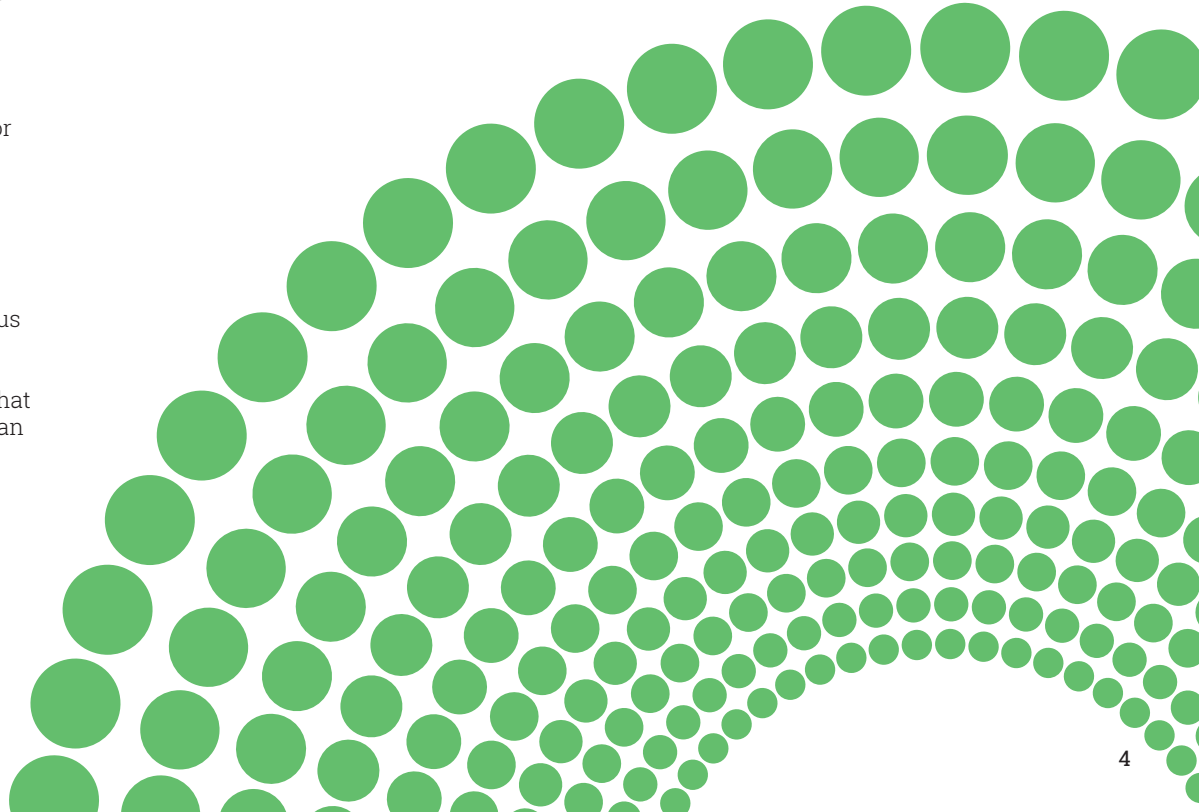
Software remote management techniques can also establish a direct connection between a vendor and the end-user of an asset, which better enables the vendor to use third party distribution channels either to enter new geographic markets or to extend reach into existing markets. In such cases, the vendor is well positioned to use a direct connection to an asset to ensure that on-board software is up to date and that the machine is functioning as it should. Usage information can also potentially highlight any training requirements that there may be, either for end-users or sales channel staff. In short, the use of remote software management allows a vendor to maintain a close relationship with an end-user, even when distribution or channel partners may intermediate in the supply chain.

Closely managing on-board software from a remote location can also help to ensure that appropriate security standards are maintained by ensuring that any required software patches are quickly and correctly applied and can help to ensure that equipment is operating as efficiently and effectively as possible. It also enables CI/CD approaches to be applied to sophisticated industrial assets, particularly including the administration, updating and ongoing refinement of any on-board AI capabilities.

Another significant benefit of remote software management is that it can help to maintain software version consistency across an equipment manufacturer's device estate, which may be distributed and installed at many widely dispersed locations for different end-users. This significantly simplifies any processes related to customer support or troubleshooting for several reasons. Firstly, any customer support agent can be certain of the software that is installed on a given device and can check that all relevant software patches and updates have been applied. Secondly, any known problems that are associated with previous software versions will either not arise (if software updates have been applied) or can be quickly dealt with by upgrading software. Thirdly, since software versions can be managed in a relatively homogenous way across an entire estate of assets installed in different customer locations, so the technical problems that are likely to emerge will be less diverse and the task of providing technical support is simpler and requires less training: it is far less likely that issues associated with legacy software versions will arise, and if they do then they can be remotely addressed with a software update.

An associated benefit is the ability for a vendor the ensure compliance with licence terms, including the potential to remotely disable assets if licence terms expire, or if all available licences have been allocated.

There are also downstream benefits, wherein closer monitoring of the use of assets on client sites can extend the relationship between a vendor and an end-user adopter, both over a longer period of time and also in terms of the depth of the relationship. The information that a vendor can collect from assets in the field can provide insight into when an individual customer might benefit from an upgraded service and position the vendor to offer that upgrade as part of an ongoing service and via existing channels of communication. Aggregated information from assets deployed with a range of customers can provide input to the development of future products and services. Vendors can also potentially commit to higher service levels, based on closer monitoring of assets, and extend propositions to include concepts such as business continuity insurance to compensate for any downstream effects of service outages.



The role of a software monetisation platform

The benefits of remote software management are clear, but it is also a potentially very complex area. As such, specialised solutions have emerged to help vendors to manage software that may be deployed on remote assets. Such solutions will ideally work on the basis of a standard software environment (or software container) deployed onto assets, working in tandem with a robust, flexible and auditable cloud platform. Connections between existing enterprise systems that a vendor may have and the software monetisation platform should be API-based in order to minimise the development work required to implement remote software management and maximise the impact of the new capabilities: ideally the process from customer quote through to software deployment and invoicing will be completely automated, including pushing any new software (or licence information) to an asset.

A software-centric future

The concept of software monetisation supports the adoption of new commercial models and new layers of value that can be added to the provision of assets. It also provides an opportunity for new kinds of vendor to intermediate in established value chains, including vendors that specialise in purchasing assets outright and offering those assets to end-users alongside sophisticated software-centric propositions and also vendors that offer software-enabled services to better support the provision of maintenance and licencing for assets that have been purchased outright, or even leased from a third party.