<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>Overview</td>
<td>3</td>
</tr>
<tr>
<td>TCO Analysis of Enforcement</td>
<td>4</td>
</tr>
<tr>
<td>The Development and Productivity Impact of Build vs Buy Enforcement</td>
<td>4</td>
</tr>
<tr>
<td>The Opportunity Cost of Build versus Buy Enforcement</td>
<td>6</td>
</tr>
<tr>
<td>TCO of the Back-Office: Bottom-line Expenses and Top-line Impact</td>
<td>7</td>
</tr>
<tr>
<td>Accurately Estimating the CAPEX of Building</td>
<td>7</td>
</tr>
<tr>
<td>Accurately Estimating the OPEX of Homegrown Systems</td>
<td>9</td>
</tr>
<tr>
<td>Intangible Benefits from Commercial Monetization Systems</td>
<td>10</td>
</tr>
<tr>
<td>Putting it all Together - The Return on Investment from Commercial SLM</td>
<td>11</td>
</tr>
<tr>
<td>ROI on Back-office Licensing for Mid-sized Businesses</td>
<td>12</td>
</tr>
<tr>
<td>Why Mid-sized Businesses Consider In-house Development</td>
<td>12</td>
</tr>
<tr>
<td>Risks Specific to Mid-size Businesses</td>
<td>12</td>
</tr>
<tr>
<td>When Homegrown Might Make Sense</td>
<td>13</td>
</tr>
<tr>
<td>ROI on Back-office Licensing for Large Enterprises</td>
<td>13</td>
</tr>
<tr>
<td>Why Large Enterprises Consider In-house Development</td>
<td>13</td>
</tr>
<tr>
<td>Risks Specific to Enterprises</td>
<td>13</td>
</tr>
<tr>
<td>When Homegrown Might Make Sense</td>
<td>14</td>
</tr>
<tr>
<td>Solution Brief: Gemalto</td>
<td>14</td>
</tr>
<tr>
<td>Conclusion</td>
<td>15</td>
</tr>
</tbody>
</table>
INTRODUCTION

Software licensing and monetization (SLM) systems are designed to safeguard software-powered products from rampant piracy and counterfeiting, and streamline the business transaction process. The profitability and growth of software businesses hinge significantly on getting the monetization components right. However, too many software publishers believe that an investment in software management systems is an option rather than a business-critical imperative. This is because under-licensing of software or sub-optimal back-office operations do not manifest as explicit bottom-line expenses, while the costs of commercial SLM systems are real and tangible. Frost & Sullivan research finds that misplaced mindsets regarding the high perceived cost of a commercial licensing system and the low perceived complexity of building it in-house are leading nearly one of two software companies to choose to build instead of buy their SLM system.

This approach may be penny wise, but it is pound foolish. If you are an ISV or an embedded product vendor, you are in business because you excel at building your specific product. Undertaking in-house development of a full-fledged monetization solution is equivalent to instituting a new product line which is tangential to your intended product roadmap, outside your core competency and a drag on your bottom line. Spending scarce resources in an attempt to experiment in reinventing the wheel is a risky business strategy. Ask yourself the question: Can you truly spare the resources to build and constantly evolve your monetization layer along with your business? Software developers are continuously grappling with new technologies, features and use cases in the context of their own core product lines, and falling short of resources to do so. Can you really afford to support the ongoing investments to your monetization layer that are crucial to ensuring your business can stay on top of new business models, support new computing platforms, enable new usage scenarios and combat the latest hacking technologies?

The argument to buy an SLM system instead of building it is straightforward from the core competency and business agility point of view. We find that it is clearly justified from a total cost of ownership (TCO) perspective as well. Moreover, the TCO emerges in favor of buy instead of build, not just for large enterprises, but for any size and any type of business. This paper provides a deep-dive, objective analysis of the technical and business aspects that should inform a build-versus-buy decision for software monetization. This framework can be used to evaluate various software monetization product alternatives to choose the one that is best suited to deliver optimal TCO to your business.

OVERVIEW

Two distinct components make up an end-to-end SLM system. The first is technical enforcement used to tackle tampering and piracy. The second is a back-office system designed to issue licenses, facilitate enforcement, and deliver metrics to various internal stakeholders. Together, these components enable revenue generation, increase operational efficiency, preserve a company’s bottom line, and provide intelligence to fuel future growth. They also help curb unauthorized use, and help enable legal measures or business remediation where necessary.

Enforcement is the front end of an SLM system and is a security-centric component. It can be achieved electronically or with the assistance of hardware. Upfront licensing costs for enforcement alone tend to be modest, with the majority of costs typically incurred incrementally as licenses are issued. The enforcement-end typically needs to be tightly integrated with the software or software-powered product, such that switching vendors or license-definition schemas is difficult, although it is still not prohibitively expensive. On the other hand, if there is a failure of security in the enforcement layer, such that enforcement tokens can be freely replicated or simple hacks can be used to turn off or bypass the enforcement, then monetary repercussions are severe and mitigation is difficult.
The back-office component is far more extensive and accordingly is much more expensive to build; it is also complex to maintain. Because it is closely intertwined with business operations infrastructure, it is typically very expensive to switch out or transition away from. That said, a company's agility with business models, operational efficiencies through automation, business optimization leveraging metrics, and a host of other tasks crucial to preserving bottom-line costs and top-line revenues are directly dependent on back-office efficiency, agility and scalability.

In this paper, we will take a detailed look at the TCO for each component. TCO consists of two components—upfront Capital Expenditure (CAPEX) and ongoing Operational Expense (OPEX) to run, maintain and upgrade the system. CAPEX is commonly mistaken to include only development costs, but in reality there are several aspects to CAPEX cost, including the planning, development, deployment and testing of enforcement and back-office components. It is also critically important to realize that monetization is a program, not a project, and needs to be budgeted for and resourced as such. Accordingly, companies must account for ongoing upgrade and expansion costs as part of ongoing and continuous OPEX.

We will first dive into the considerations for modeling TCO for enforcement, followed by an analysis of the TCO of the back-office layer.

TCO ANALYSIS OF ENFORCEMENT

Aspects of CAPEX and OPEX for enforcement include not only the upfront R&D effort, but also the ongoing cost of maintaining the technology. Companies must realize that on an ongoing basis, engineering resources are required to update the enforcement layer, test it and distribute it quickly to clients—often at the expense of planned product upgrades, thus causing roadmap disruption. It is in breaking this recurring attack-mitigate-attack cycle that the true value of a commercial solution manifests itself, as we will see in the sections below.

The Development and Productivity Impact of Build vs. Buy Enforcement

The diagram below shows the typical development pattern for an enforcement component built in-house. The initial release of the technology is usually a planned R&D exercise, with adequate resourcing and streamlined release management. Companies will often consider only the initial development cost of the component and conclude that building it in-house can be feasible as well as economical. At the time of release, the strength of the enforcement achieved is usually adequate to mitigate the risk of piracy and counterfeiting to a degree that is appropriate for the company’s risk profile as interpreted by its engineering team. Over time, however, this best-effort engineered component becomes the target of attack groups and its effective strength begins to fall, leading to a rising level of risk. As indicated in the figure, the frequency of hacks typically rises in correlation to the risk level.
The cause of ongoing piracy challenges is that on an ongoing basis, most in-house engineering teams are concerned with roadmap features related to the core product, with lower priority accorded to monetization aspects. When attacks inevitably emerge sooner than expected, a fire drill typically ensues to scramble up resources and patch the vulnerabilities. Furthermore, because this patching is done under schedule and resource duress, the restored level of security is inevitably lower than ideal, resulting in a progressively accelerated hack cycle and increasing frequency of engineering disruption. Within a couple of cycles, companies nearly always reconsider the decision to build in-house and begin to evaluate commercial alternatives. However, both time and dollars have been wasted in the meanwhile. The opportunity cost—from lost resources that could have been used to enhance core product lines, and from lost revenue resulting from delayed piracy management—is also considerable.

Building licensing components in-house forces your product manager to be a licensing expert and security expert as well; in most cases, we find that this is an unrealistic expectation.

—David DiMillo, Principal Consultant, Software Monetization, Gemalto

The basic issue with building enforcement in-house is that you are forcing your product management, development and testing teams to develop a competency in security and monetization, in addition to their core competencies around your product itself. In contrast, judiciously choosing a monetization partner and relying on a commercial monetization solution offers predictable update schedules, more durable protection against hacks, and the ability to maximize focus on product-related development and innovation without compromising on revenue generation. The implications of these factors are shown in the figure below.
With any high-quality commercial solution, the risk profile remains fairly stable, given that the vendor will proactively be upgrading its security features as hacking state of the art advances. Thus, by the time hacks emerge, a new version of security is typically well under way. Another source of hacks can be errors in integrating the solution into the software product. Either way, fixing these issues is a relatively minor R&D exercise as compared to the effort involved in re-crafting an enforcement component from scratch. Moreover, because these updates can be executed in planned, scheduled fashion rather than as a disruptive fire drill, they do not adversely impact product schedules or stress resources. Finally, because hacks stay controlled and never reach cascading volumes, the risk of massively increased revenue losses as a result of such hacks remains minimal.

An ISV’s partnership with its monetization vendor needs to go beyond just a solution sale. For example, if a hack should occur, the partner should also have reliable services to help determine how you were hacked and how to fix it. Having an expert pool of advice and resources is crucial to a business whose profitability hinges on effectively monetizing its product.

**The Opportunity Cost of Build versus Buy Enforcement**

Frost & Sullivan sees software monetization not simply as a revenue protection solution, but more fundamentally as a business enablement technology. Accordingly, your monetization layer needs to scale and adapt as software consumption models change and SKU form factors change—just as your product portfolio does. If the monetization layer cannot—or does not—adapt with the same agility, then it holds back progress, negatively impacts competitive positioning and hinders revenue growth.
Opportunity cost factors such as lack of flexibility in licensing models, taking developers away from product engineering, and reduced agility in the cloud are difficult to quantify but can tangibly hold a company back in its quest for innovation and growth.

—Avni Rambhia, Industry Principal, Frost & Sullivan

What is revolutionary with an in-house implementation can become simply evolutionary with a reliable partner. This is an important aspect to the value a commercial solution can provide.

There can be many opinions on how to accurately model the opportunity cost imposed by in-house monetization solutions that hold back flexibility in deploying new business models like pay per use or expanding support to new platforms like tablets. Given a choice in allocating scarce resources, would you rather push the bar in their core competency, or dabble and experiment on the monetization front? At the end of the day, the strongest argument in favor of a business of any size to consider a commercial monetization solution is that it allows your product management, development, testing and IT teams to focus on their core competencies, while relying on the expertise and foresight of a trusted partner to ensure the monetization layer can keep pace with changing use cases, business models and computing landscapes.

When we work with our customers, we’re building a trusted advisory and partnership relationship with them. They can take advantage of our 30 years of understanding of monetization requirements, rather than reinventing the wheel alone. With Safenet, ISVs effectively get a product architected based on the requirements of thousands of customers, and thus can be sure that the monetization layer will support the business long term.

—David DiMillo, Principal Consultant, Software Monetization, Gemalto

TCO OF THE BACK OFFICE: BOTTOM-LINE EXPENSES AND TOP-LINE IMPACT

Accurately Estimating the CAPEX of Building

Frost & Sullivan finds that many companies, when evaluating the costs of building a back-office component in-house, make the error of only considering cursory design and baseline implementation costs. In reality, for the back-office layer to run smoothly and be scalable and agile over time, you also need to factor in the cost and schedule of full-fledged planning, design, implementation, functional testing, documentation and security-focused penetration testing. The figure below shows this fully loaded cost of a build exercise, relative to the typical initial estimates of cost.
Moreover, in-house projects will nearly always overrun both schedule and budget, while also falling short on functionality. The figure above shows the implications of accounting for this level of risk in a CAPEX estimate for build. Industry-average statistics for in-house software development indicate a 60% risk of failure at first attempt. If we seek to reduce that risk to a more palatable 10%, the corresponding CAPEX estimate rises by a factor of 2.5. When comparing this far more realistic cost estimate to the cost of a commercial solution, the business case emerges in favor of buy instead of build. Notably, this relative cost estimate is as true for medium-sized businesses as it is for large enterprises. While the actual dollar levels vary by size of company and complexity of deployment, the relative costs work out to be similar in all use cases except for very small (<$3 million/year in revenue or <500 units/year), fairly simple deployments.

Figure 2: Ongoing Cost Projections for In-House Monetization Systems
Note that these numbers do not account for any opportunity costs or top-line savings generated by the shorter timeline of deploying a commercial solution as compared to a homegrown system. Typically, homegrown systems take anywhere from three months to one year in development, depending on complexity and in-house resource availability. In contrast, commercial solutions are typically deployed over four to six months, depending on complexity and scale. The savings in terms of calendar month duration of the project and resource man-months are shown in the table below for small to medium-sized businesses, and for large enterprises. For SMBs, the average resource savings are the key metric. For example, an SMB with a team of 20 engineers saves nine man-months. This is equivalent to approximately 5% of resource availability. On the other hand, a large enterprise may have a few hundred resources and therefore 26 man-months is more or less a rounding error, but the loss of seven calendar months prior to roll out can correspond to tens of millions of dollars in compromised revenue or reduced sales, if not more.

<table>
<thead>
<tr>
<th>Type of Company</th>
<th>Average Shortening of Project Duration (Buy versus Build)</th>
<th>Average Resource Savings (Buy versus Build)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMB</td>
<td>4 months</td>
<td>9 man-months</td>
</tr>
<tr>
<td>Large Enterprise</td>
<td>7 months</td>
<td>26 man-months</td>
</tr>
</tbody>
</table>

Software licensing touches almost every department throughout our company, from our content department to operations, IT, and accounting. We needed a licensing system that would bridge our back-office CRM and ERP systems to provide quick updates to the end user, said José Ignacio Martínez Colomer, AEOL Managing Director and Founding Partner.

**Accurately Estimating the OPEX of Homegrown Systems**

There are three categories of ongoing costs that companies should account for when estimating the actual OPEX of their monetization system.

- **Basic Ongoing Cost Factors:** Companies will typically spend two to three man-months a year on routine maintenance or internal technical support issues, such as bug fixing, expanding system capacity, making allowances for new product releases or new product versions, updating systems as necessary to account for the latest operating system versions and patches, and keeping servers up and running. Also significant are routine external-facing tasks such as technical support, which typically costs about $140 per hour per call. It should be noted that the average number of calls as a ratio to total customers tends to be higher for in-house architected solutions as compared to when a professional solution is being leveraged.
• **Enhancement Cost Factors:** Enhancements are major functional upgrades to the monetization system that improve business efficiency and maximize ROI from the system. Most often, in-house teams will tend to leave value-add features like these for a Phase II or subsequent update, but lack of expertise/confidence and paucity of resources will often push these further back. Common examples of such features are visualization and metrics reporting for non-engineering teams such as finance and marketing; automation of license issuance, auditing and billing; metrics and analytics; and globalization/localization features. With well-architected systems, companies will typically spend 10-20% of original build cost in enhancement.

> We often find that for homegrown systems, Phase II essentially becomes Phase Never.  
> Avni Rambhia, Principal Analyst, Frost & Sullivan

• **Rewriting Costs:** Many companies with in-house systems report that every two to four years they outgrow their current systems and face a complete overhaul effort to support current customer use cases and business needs. This will typically carry a cost overhead at least equal to the upfront CAPEX, if not higher, if it is to be properly addressed. Rewriting costs will vary widely depending on how future-proof, visionary and well documented (or not) the original deployment was.

### Intangible Benefits from Commercial Monetization Systems

It is very tempting, from an analyst perspective, to measure ROI as not only the bottom-line savings, but also take into account the top-line revenue increases enabled by effective leverage of the monetization system. The problem is that while this approach is reasonable, the resulting numbers will inevitably indicate that monetization solutions will pay for themselves, often several times over, within a year. Such statistics can be hard to take seriously, even if they are justifiable. Accordingly, we will not account for top-line benefits in our numerical ROI analyses below. Nonetheless, we capture them here for the sake of completeness and to provide a framework for business leaders or product managers to work them into their own models as they feel appropriate for their specific business.

• **Metrics-driven sales growth:** High-end monetization solutions will offer the ability to gather statistics and report metrics on which customers are using which features most often, allowing a data-driven marketing team to tailor upgrade and upsell proposals and thereby grow revenue. For traditional on-premises software, this benefit usually ramps up from 5% of annual revenue in year one to as much as 25% of annual revenue by year five.

• **Agile adoption to new models:** The ability to quickly expand software utilities (whether standalone software packages or software extensions of embedded devices) to mobile and SaaS use cases can yield immediate new upsell revenue and can also help get ahead of competition. On the flip side, lack of support for a critical range of portable devices can result in lost sales and long-term growth impact. The quantification of this factor varies significantly across businesses, but we have seen SaaS revenues for agile companies grow by as much as 300% within a year and 500% over five years, allowing growth in stagnant markets and yielding a very effective global expansion strategy.
• **Reduction of losses from piracy and counterfeiting:** In terms of customer numbers, approximately one in two enterprises is likely to be (intentionally or in error) overusing licensed software. In terms of revenue dollars, the norm for the software industry is to assume a 15-20% loss of total revenue to piracy and under-reporting. On average, we find that adopting a monetization solution improves total revenues by 3-7% over the course of two years. In other words, effective enforcement can reduce the piracy/overuse rate by 25%. Counterfeiting is a far more serious situation, with a potential loss of revenue that can equal or even significantly exceed actual generated revenues in a given year. A 2015 survey of ISVs by VansonBourne found that over four-fifths of ISV respondents are worried that their software products might become compromised. Monetization technology on its own cannot tackle counterfeiting, but it is a key component of an anti-counterfeiting strategy and as such plays a crucial role in mitigating business risk.

• **Automation:** As businesses scale upward and outward, automation of license issuance, product upgrades, marketing reports and compliance reporting—to name a few—all result in improved performance, better sales, better customer satisfaction and improved resource utilization. Automation benefits can usually reduce the ongoing operational expense by a quarter to a third.

• **Better compliance and lowered auditing costs:** Under-reporting is among the easiest problems for software vendors to solve, assuming a reliable way to measure usage data, report it back, and correlate it to licensing information. It is fair to assume that all under-reporting will be discoverable by the software publisher within two years of deploying an effective enforcement and analytics solution. Moreover, reliable data makes the true-up process far simpler, with lessened need for expensive manual audits and fewer unpleasant client conversations. The net benefit (top-line growth from true-up licensing and bottom-line savings from reduced auditing) from this factor can range from 5 to 33% of annual revenues, depending on the type of software product and prevalence of under-licensing with the current monetization solution.

**PUTTING IT ALL TOGETHER - THE RETURN ON INVESTMENT FROM COMMERCIAL SLM**

The graph below shows the relative CAPEX and average OPEX over four years for back-office build and buy scenarios for a typical medium-sized company. (Note that we consider neither costs nor benefits of commercial enforcement in these models). The key takeaway is that the buy scenario offers predictable and steady OPEX; we will see in the next set of graphs how this delivers measurable long-term benefits.
The graph below shows the five-year cost of ownership of the back office in build and buy scenarios on a net present value basis, as a percentage of company revenue. It also shows the cost of a commercial solution adjusted very nominally for a 3% increase in monetization efficiency, as a very conservative indication of potential return on investment. We see that over the long term, the raw costs of commercial and homegrown solutions become comparable as company size and licensing complexity increase. Once even a minimal increase in monetization efficiency is factored in, it becomes apparent that a commercial solution will not only be cheaper, but will decisively pay for itself—certainly for the larger enterprises, but even for medium-sized businesses.

**Figure 5: Typical-Year TCO of a Back-Office System, by Company Size, as a Percentage of Company Revenue**

**ROI on Back-office Licensing for Mid-sized Businesses**

**Why Mid-sized Businesses Consider In-house Development**

Mid-sized businesses are most likely to consider in-house development of software monetization due to perceived cost. Such businesses typically ship fewer units of product per year, but the upfront cost of a commercial back-office component is not dependent on revenue or quantity of product shipped, so their perceived per-unit cost of a commercial solution is higher. Moreover, the integration fees are the same regardless of size of company or product revenue, so the CAPEX can seem overwhelming for these businesses. Furthermore, the can-do engineering mentality that is usually a great strength for these businesses, combined with the budget pressure that invariably plagues smaller companies, collectively tend to result in enthusiasm for tackling the problem internally.

**Risks Specific to Mid-size Businesses**

Mid-sized businesses are very likely to have ad-hoc development and documentation procedures, particularly those still in start-up or aggressive growth mode. The biggest risk in the long term is that of “tribal knowledge.” If the key developer switches jobs or does not move when the company relocates (as examples), and there is not enough documentation (as is typically the case), then maintaining and improving the monetization components becomes very difficult and a ground-up re-engineering effort on a fire-drill basis can often become unavoidable. Additionally, limited time and resource availability nearly always drives feature decimation, with many key features offloaded to a potential Phase II (also known as Phase Never). Customer service overhead
from glitches in the enforcement or license distribution components can result in debilitating volumes of technical support calls. Finally, companies find that as order volumes begin to exceed the volume of 10 or 15 per day, the overhead of entering orders and manually issuing licenses (if this is required by the homegrown system) is simply too much to manage without dedicated resources, at which point OPEX considerations erase any potential savings from homegrown development.

When Homegrown Might Make Sense

There are some cases where building a homegrown system may ultimately make sense for a medium-sized business. We find this is often true for a company with fewer than 500 unit sales per year (i.e., less than 10 per week), or under USD 3 million in revenue, whose licensing requirements are fairly straightforward (e.g., a product is either enabled or it is not, with no feature-based licensing or need for floating licenses). In these cases, it can make sense to temporarily use an in-house solution as a stop-gap measure until business size increases. Outside of these fringe use cases, however, we find that there are real savings to be gained from switching to a commercial solution when the total cost of ownership as extensively described above is considered.

ROI on Back-office Licensing for Large Enterprises

Why Large Enterprises Consider In-house Development

Large companies are more sophisticated in build-versus-buy decisions, have experience managing large budgets, and are likely to take a long-term cost of ownership approach by default. However, enterprises typically have in-house IT teams who offer de-facto professional consulting services, and so the choice is often between buying internally from the IT team versus buying externally from a software monetization vendor. Until the early 2000s, lack of vendor maturity resulted in a dearth of full-featured commercial back-office solutions, which has contributed to the prevalent mindset that commercial solutions are neither available nor adequate for large-scale enterprise back-office needs.

“

A license management system is not entirely different in its scope from ERP or CRM systems. The most significant similarity is how they all touch almost every department in an organization. This is where an in-house system can become problematic.

—David DiMillo, Principal Consultant, Software Monetization, Gemalto

“

Risks Specific to Enterprises

As attractive as the notion of utilizing in-house IT resources might be, the track record of internal IT teams delivering projects on time, within budget and to specification is dismal. Industry statistics show that nearly one in three internal IT projects will fail completely and be cancelled, and another one in three will be challenged by budget and schedule overruns. The likelihood of an IT project succeeding at first attempt is under 30%, which is a dangerous level of risk for a business-critical component such as monetization. Customer research shows that over four in five large enterprises currently using in-house systems are dissatisfied with them and are likely or very likely to transition to an external firm. Additionally, high sales
volumes imply higher impact and disruption when a transition is needed, either from an old in-house solution to new in-house solution or in-house solution to external solution. Ideally, an enterprise product manager only wants to overhaul their SLM system once, and then rely on ongoing upgrades to the platform without active in-house engineering involvement.

**When Homegrown Might Make Sense**

We have seen select cases where a large enterprise with a very complex product, a very large user base, and/or extremely high internal technological competence has built out an adequately secure, adequately efficient and adequately scalable monetization system in-house. There is also a small community of enterprise users who leverage commercial monetization but an in-house back office and are satisfied with this set up. Characteristics distinguishing these enterprises include very specific requirements, stable business models, an adequate pool of trained resources, and adequate business awareness and motivation to fully invest in monetization as an ongoing business-critical program versus a one-off project. With these characteristics, the risk levels of in-house development drop to manageable levels and the ongoing need for enhancements and re-engineering effort is low. In all other cases, the economic model below indicates the benefits that can be realized by leveraging a commercial monetization solution.

> A licensing system is not a project; it is a program. Projects have finite starts and ends. If you look at this as a project, it will fail. This is a program; it needs to be nurtured, tweaked and evolved over time. It has to scale; it has to be owned.
> The companies who are most successful—build or buy—are the ones who treat monetization like a program.
> —Ava Diamond, Sr. Manager, Product Marketing, Gemalto

**SOLUTION BRIEF: GEMALTO**

Gemalto (Safenet/Rainbow Technologies) is distinguished by its long years of experience in implementing software monetization for a wide range of businesses and use cases. The company provides software, hardware and cloud-based enforcement and entitlement solutions; SaaS-based back office; and on-premises back-office options. A wide spectrum of product form factors, from embedded to desktop to cloud, is supported. Recognizing that a large percentage of the developer community is transitioning from in-house to commercial monetization solutions, and that this transition is both expensive and disruptive, Gemalto consciously takes the approach of enabling the transition from homegrown to commercial model in a gradual evolutionary manner as opposed to forcing a watershed revolutionary shift. Gemalto positions itself as a security partner who insures your business against ongoing advances in hacking technology, as well as ongoing changes in the software and software-powered business environment. Further bolstering this partnership approach to monetization is its professional services organization, which offers a full spectrum of services from business planning to customization and deployment to risk management, building on over 30 years of experience in knowing what software-powered businesses need to be successful and agile.
Frost & Sullivan research found that Gemalto has been the leading vendor by revenue market share in the software monetization market for three years in a row, owing to its outstanding product line strategy, its business-enabling partnership approach, its modern and agile product features, and its broad geographic footprint.

CONCLUSION

Monetization is a program, not a project, and needs to be budgeted for and resourced as such. Despite being the nerve center of a software company’s profitability, monetization is too often treated as a necessary cash drain, a stepchild project or a one-step checkbox. As a forward-looking business leader or a savvy product manager, how do you make the convincing argument for this to be a high-priority, long-term corporate program? This paper has provided a comprehensive framework for determining the TCO of build compared to buy software monetization solutions based on quantifiable bottom-line costs and justifiable top-line considerations.

It is important to realize that homegrown systems will have to go through a radical re-engineering every four to five years, with an inevitable recurring CAPEX spike. In contrast, a high-quality commercial solution will constantly upgrade those features for you, so you switch to a predictable OPEX model right from year one. Even if significant upgrades are needed as the business evolves (e.g., when expanding into cloud-based options to supplement on-premises licensing, or switching to a pay-per-use business model), a commercial solution will allow an evolutionary cost structure, whereas an in-house architecture will force a watershed disruption.

With a commercial solution, it is true that upfront costs will be moderately higher than the homegrown alternative. However, the lowered risk and deterministic long-term returns are well worth it. It’s unlikely you would consider building your new office headquarters yourself; you take a long-term perspective and invest in expert contractors. The same analogy holds true for software monetization. You need to focus on doing what you are good at, come out with great software, and leave the enforcement and back office to monetization vendors who know what they are doing. You will find yourself with improved business agility, increased productivity and clear operational visibility with lower long-term expense.
Frost & Sullivan, the Growth Partnership Company, works in collaboration with clients to leverage visionary innovation that addresses the global challenges and related growth opportunities that will make or break today’s market participants. For more than 50 years, we have been developing growth strategies for the Global 1000, emerging businesses, the public sector and the investment community. Is your organization prepared for the next profound wave of industry convergence, disruptive technologies, increasing competitive intensity, Mega Trends, breakthrough best practices, changing customer dynamics and emerging economies?

For information regarding permission, write:
Frost & Sullivan
331 E. Evelyn Ave., Suite 100
Mountain View, CA 94041