Three key requirements for Generative AI monetization

An Intellyx Analyst Guide for Amberflō

Featuring thought leadership from Jason Bloomberg, Jason English and Eric Newcomer

Table of Contents

Introduction

There's a virtual gold rush going on, as technology investors and vendors are making huge bets on tying Generative AI-powered features to virtually every application we use today.

But once extensive resources are committed toward developing an AI-powered service, how can all of that money and energy get converted into real business revenue and deliver measurable value to customers?

This Intellyx guide, sponsored by Amberflo, will help readers understand the three key requirements for monetizing Gen AI through better customer acquisition, billing and retention, while tracking business performance and costs.

By Jason English

Director & Principal Analyst
Intellyx

Monetizing generative AI starts with metering

Part 1 of the Requirements for Generative AI Monetization Series

It seems like every vendor is pushing an AI-washed story these days. Generative AI hype is at an all-time high thanks to ChatGPT, Midjourney, and many newer startups and projects. Companies that fear missing out on the next big thing are spending money on AI functionality as a loss leader or slapping an LLM chatbot in front of their apps.

It makes me wonder: *Wait a minute, is this really going to help customers? And, how are we supposed to start making money from so much AI hype?*

For most companies, building out a complete AI stack would be prohibitively expensive and resource intensive. Who has the money and resources to literally design their own AI, and how would the new venture make enough money to stay afloat, considering massive R&D and infrastructure costs?

Overinvestment in AI is not the answer

Right now, venture capital is on the hunt for any startup claiming to revolutionize machine learning or use GenAI in some novel way. Tech titans are placing big bets on tools in the sector. Will another project even come close to getting $10 billion dollars for a minority share like OpenAI got from Microsoft?

You never know what might happen up there in the stratosphere. For the rest of our efforts here on Earth, we can run experiments, but we will only be able to meaningfully adopt AI once it provides business value for customers.

Eventually the bills will come due on so many AI projects, and there will be a great reckoning that will divide the products that can find a market niche and help companies capture revenue, from those that won't.

The cost of infrastructure will get really high, really fast, as will the cost of GPUs, data ingress/egress, and architecture. Companies large and small will struggle finding any experienced AI modeling experts or machine learning data scientists willing to start working on yet another project.

A supply chain of loosely coupled composite AI applications

If enterprises want to see a return on their AI investments, they must prove that their chosen strategy is applicable to real-world business and societal problems, rather than serving as window dressing.

Since we can't depend on just one form of AI, to get there, we'll need ***composite AI*** – a supply chain consisting of multiple AI suppliers, with multiple models and multiple training data sets working together in a loosely-coupled fashion, based on the right fit for the job.

Much like any other industry, from producing cars to sneakers, manufacturing will collaborate with specialized supply chain and logistics partners to fulfill customer demand—there will be a supply chain for assembling composite AI-based applications from model components.

AI producers and consumers need a multi-party interface to flexibly associate with each other. Since each provider would need to interface with others, the interface between partners would likely be expressed as an API, but with specifications for the workload at hand.

Metering is the dialtone of composite AI apps

Just like any SaaS platform or app on your smartphone, when you use AI-driven applications, you are entering into a partnership with the vendor as well as underlying AI providers—though the contracts may be represented as APIs rather than formal terms of service.

How can we manage the ground rules of AI services so everyone gets paid enough to continue improving quality and performance, while still delivering for customers without surprise billing or functional failures? It's a huge problem, and current approaches leave the wires hanging, with real customer service and liability implications on the line.

The AI supplier pays for the non-trivial initial cost of curating ML data, training the model, and making it ready for consumption through an API or an interface.

AI suppliers that don't bill for usage would still need to know what consumers are doing with their AI-based application—even if the only point of doing so would be to justify the investment for community utility and business partners.

We can solve this problem with better AI metering, kind of like we used to do with telco providers for decades. The party on either end of a call would buy their own local telephone infrastructure with flat rates for local calls, while the long distance or international calling service would handle the remote connection on their own network, billing the sender or receiver of the call by the connection, or by the minute.

AI inference models, LLMs and feature libraries are hosted in data centers and cloud hyperscalers around the world, as reserved instances or ephemeral microservices and Lambda functions. Wherever they reside, AI metering acts like a common 'dialtone' for AI-enabled applications to measure, assemble and run workloads.

Bringing AI metering out for distribution

Major cloud marketplace aggregators like Salesforce, Azure and AWS are well known for offering huge catalogs of subscription-based application services and tools, including AI services, but the measurement of usage and billing generally occurs within the confines of that aggregator's account interface.

Enterprises that deliver applications have gotten used to applying various forms of observability and monitoring to try and understand usage, but once logs are recorded, they are already history.

To build composite AI apps, we need a way to understand exactly how a mix of models are invoked and executed to contribute to each end result.

Take for instance a financial advisory composite AI app. When an investment client asks for some recommendations, the advisor asks the app through a conversational LLM for value predictions based on 6, 12, and 24-month horizons. The LLM then invokes one AI service for risk profiling, another for comparing asset categories, and yet another service that looks at trendlines. Still another GenAI tool may generate a client report out of all of the results.

Each model invocation in the chain is metered, so the company can determine which models are proving valuable to customers, and how to compensate all of the providers or check their work.

Metering of AI models doesn't have to work any differently than metering API-based services. That's why **Amberflo** offers a Github metering repo of useful scripts and metering SDKs for modern development platforms like Go, Python and Java, and plugins for API gateways such as Kong.

The Intellyx Take

Metering is the telemetry of AI monetization and utility. It gives us insight into exactly how substrate algorithms, inference models and applied processes contribute value to composite AI applications.

Think about the alternative–***not metering AI*** would leave all providers and consumers helpless, in a multi-party chain dependent on logs and other artifacts and no common understanding of usage.

Therefore, metering should be embedded within every API service producer interface and REST-style call to a feature, as well as within the consuming AI-driven application itself.

Capturing real-time and historical usage metering will make AI a valuable contributor to critical applications, instead of a cost sinkhole.

This reactive process is so ingrained in the way operators think about incidents that entire product categories have grown around it, from traditional IT incident management to AIOps to observability.

By Eric Newcomer

CTO & Principal Analyst
Intellyx

Usage-Based Pricing Models for Gen AI

Part 2 of the Requirements for Generative AI Monetization Series

Everyone's rushing to deliver generative AI based applications and products to market and cash in on the latest trend or gain competitive advantage.

At the same time, it's not clear that everyone understands the right way to set prices to ensure a return on investment.

If you are rolling out a generative AI-based application, incorporating AI into an existing application, or creating a new product that incorporates AI, taking a few minutes to develop the right pricing model will be an essential part of determining your success.

The pricing model for AI is important not only for the revenue it brings, but also because it has to be something customers easily understand. Finally, if your pricing model isn't competitive, your customers will go looking for alternatives. This is a tricky balance to achieve.

Setting up usage-based AI pricing

First, configure usage metering for the gen AI tool you are using, as explained in the **previous blog in the series**. For an LLM-based tool, this means capturing the number of words sent in the input prompt and the number of words received in the output response, and calculating the backend price charged by the model provider according to the per-token rates.

Popular generative AI tools such as Chat GPT, Anthropic, Google Gemini, Cohere, and Mistral charge on the aggregate token (i.e. word) level over a period of time, with varying price tiers depending on the quality of service you select or on an API call rate limit.

So you first need to analyze and understand the pricing model for the gen AI tool you're using, establish the metering, take into account any variations imposed by the particular AI tool, and structure your pricing model accordingly.

It's also important to figure out the right margin to add, if any, to cover your costs and generate revenue.

If the application is internal to your organization, there's probably no need to include margin—especially if you are just passing along or dividing up the cost across different departments.

However if you are reselling the service and adding value to it, such as security scanning for the prompts and responses or training the LLMs for industry specific needs, it is standard practice to add a reasonable margin to the gen AI usage fee to account for the additional value-add and generate a reasonable profit.

Billing should be transparent, understandable, and flexible enough to adapt to the customer's current payment processes, if possible.

Finally, set up an invoicing process to submit bills to your customers and track payments against them.

Subscription-based pricing models are dying out

Many current and traditional software products rely on perpetual licenses or subscription support pricing models.

However, as products move to the cloud, which basically rents capacity across a pooled infrastructure, pricing models are changing to keep pace.

Cloud infrastructure servers fail continuously, and other computers take over application workloads dynamically. Loads are automatically scaled up and down across multiple servers. This makes it very difficult to measure or enforce a perpetual or subscription-based license, which typically counts servers or cores.

In the cloud you instead measure how a managed service such as compute, storage, or messaging consumes resources such as CPU, memory, disk, or network capacity and charge for that.

And of course, ML and AI applications are among those primarily based in the cloud. So the marriage of AI and usage-based pricing is understandable.

Summarizing gen AI usage based pricing

First, analyze meter data to understand usage patterns and identify suitable billable metrics.

After you collect and identify billable metrics, you can roll them into products for sale as product items composing usage-based pricing models. When usage starts, tally, rate, and invoice the billable metrics.

The right pricing model is all the more important for AI-based products, since there's so much competition and because they consume so many resources to process the prompts and deliver the results.

For example, recording a transcript and sending it to Anthropic for a summary has a certain cost basis. Understanding the cost basis is critical to determining the ROI of such a gen AI product.

Such flexibility in the pricing model is tricky to figure out — which is why platforms for building and managing usage-based pricing such as Amberflo are helpful.

The Intellyx Take

AI and ML-based applications typically run in the cloud or depend on applications running in the cloud for sufficient elastic capacity.

Calculating the ROI and collecting the right metrics for your AI-based applications can be tricky but it is nonetheless foundational for establishing the right pricing, billing, and invoicing practices.

Quickly and correctly setting and updating the pricing model for your gen AI based app or product is easier and faster if you use a dedicated AI monetization platform such as Amberflo.

Amberflo has predefined metrics, billing, and invoicing capabilities for the leading LLMs and AI chat bots. This really helps you figure out the right pricing or cost tracking strategy – whether it's internal to your organization or for sale to external customers.

By Jason Bloomberg

Managing Director & Analyst
Intellyx

Metering and Monetizing Multiple Language Models

Part 3 of the Requirements for Generative AI Monetization Series

In the first article in this series, Intellyx Principal Analyst Jason English introduced the concept of composite AI: a supply chain consisting of multiple AI suppliers, with multiple models and multiple training data sets working together in a loosely-coupled fashion, based on the right fit for the job.

He explains how metering such composite AI-based solutions is central to the monetization of such applications.

In the second article in the series, Intellyx CTO Analyst Eric Newcomer explained usage-based generative AI (genAI) pricing, and how different genAI tools each have different pricing models.

Regardless of whether a genAI-based offering leverages one or many models, monetization of any technical solution boils down to one incontrovertible fact: revenues must exceed costs.

Given how new and untested today's genAI-based solutions are, both sides of this equation can be difficult to predict as solutions scale to meet demand, especially when the solution depends upon multiple language models.

The Many Reasons to Combine Multiple Language Models

Why would an organization build a composite AI-based application in the first place? And given all the complexities of the various pricing models and usage patterns, how should organizations go about metering and monetizing multiple language models? Here are some likely scenarios.

Combining the best features of multiple publicly available large language models (LLMs) – As Eric Newcomer pointed out, each model on the market has its own strengths, based on both its training data as well as the construction of the model itself. Most of today's genAI solutions combine LLMs to leverage their respective strengths.

Price arbitrage – Some organizations shift workloads from one model to another to get the best deal at the time. However, because each model works differently, such arbitrage can be more complicated (and thus expensive) than, say, price arbitrage across public cloud providers. Such arbitrage is impossible without genAI metering.

Mixing DIY and public LLMs – An organization building its own 'do it yourself' LLM can achieve market-leading differentiation, but creating an LLM from scratch is a Herculean effort.

Even if your organization has such ambitions, it may not want to put all its eggs in the DIY model basket. Instead, combine internally constructed models with public ones to manage costs and augment the abilities of a single model.

Leveraging models of different sizes – Much of the buzz around genAI centers on LLMs – but smaller language models can provide complementary capabilities to LLMs at a lower cost.

Smaller language models are more likely to be task-specific than the more general purpose LLMs. They require fewer computational resources, making them more cost-effective and accessible, especially for DIY scenarios.

Crafting the right usage and pricing models for combinations of language models of different sizes, therefore, can be essential for delivering a cost-effective AI-based offering.

Leveraging domain-specific models – the general-purpose models like the GPT family, LlaMa, and Falcon may get most of the press, but domain-specific models that their creators have trained and tuned for industry or use case-specific purposes are coming to market all the time.

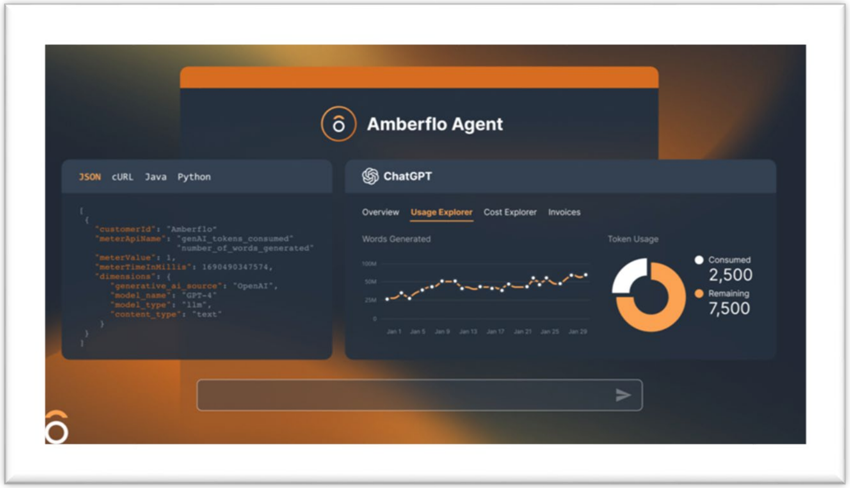
Domain-specific models are typically smaller than the general-purpose ones, and thus can be less expensive to deploy. A particular genAI solution, however, is likely to combine both types of models to deliver a differentiated capability to the market, thus requiring careful metering and billing capabilities.

The Amberflo Solution

Amberflo has updated its cloud metering and monetization solution for genAI-based applications. The platform provides usage metering, cost tracking, and customer billing capabilities for any combination of LLMs that a customer would like to deploy.

Amberflo is well-suited for such metering and monetization of multiple LLMs. With Amberflo, operators can dynamically switch models and versions of LLMs, adding custom metrics as needed.

Operators can also track per-customer, per-team, or per-user LLM usage across different models, correlating model usage costs to implement customer-friendly, yet profitable pricing models.



```
[
  {
    "customerId": "Amberflo"
    "meterApiName": "genAI_tokens_consumed"
                    "number_of_words_generated"
    "meterValue": 1,
    "meterTimeInMillis": 1690490347574,
    "dimensions": {
       "generative_ai_source": "OpenAI",
       "model_name": "GPT-4"
       "model_type": "llm",
       "content_type": "text"
     }
  }
]
```

The Intellyx Take

What all the various composite AI scenarios have in common is their complexity as compared to single model-based solutions – and with such complexity comes cost.

Given the massive data and processing requirements of LLMs, costs can easily run away from you, especially in composite scenarios. Transparent billing is equally important for complex composite AI scenarios, as your customers require control over their own costs.

While you need to pass your costs on to your customers, you must also ensure you give them value for their dollar. Without adequate usage metering and billing, you and your customers are working in the dark.

While you need to pass your costs on to your customers, you must also ensure you give them value for their dollar. Without adequate usage metering and billing, you and your customers are working in the dark.

Copyright ©2024 Intellyx B.V. Intellyx is solely responsible for the content of this guide. As of the time of writing, Amberflo is an Intellyx customer. No AI chatbots were used to write this content. Image sources: Adobe Express, and Images Money, flickr CC 2.0 license.

About the Analysts

Jason Bloomberg is Managing Director and Analyst of enterprise IT industry analysis firm Intellyx. He is a leading IT industry analyst, author, keynote speaker, and globally recognized expert on multiple disruptive trends in enterprise technology and digital transformation.

Mr. Bloomberg is the author or coauthor of five books, including *Low-Code for Dummies*, published in October 2019.

Jason "JE" English is Director & Principal Analyst at Intellyx. Drawing on expertise in designing, marketing and selling enterprise software and services, he is focused on covering how agile collaboration between customers, partners and employees accelerates innovation.

A writer and community builder with more than 25 years of experience in software dev/test, cloud and supply chain companies, JE led marketing efforts for the development, testing and virtualization software company ITKO from its bootstrap startup days, through a successful acquisition by CA in 2011. Follow him on Twitter at @bluefug.

Eric Newcomer is CTO and Principal Analyst at Intellyx, a technology analysis firm focused on enterprise digital transformation. Eric is a well-known technology writer and industry thought leader, and previously held CTO roles at WSO2 and IONA Technologies, as well as chief architect and chief security architect roles at Citigroup and Credit Suisse.

About Intellyx

Intellyx is the first and only industry analysis, advisory, and training firm focused on customer-driven, technology-empowered digital transformation for the enterprise. Covering every angle of enterprise IT from mainframes to cloud, process automation to artificial intelligence, our broad focus across technologies allows business executives and IT professionals to connect the dots on disruptive trends. Read and learn more at https://intellyx.com or follow them on Twitter at @intellyx.

About Amberflo

Amberflo enables businesses to track and charge on usage and bill customers with on-demand, accurate metered invoicing. Based out of the San Francisco Bay Area, Amberflo is supported by investors including Norwest Venture Partners, Homebrew, and Operator Collective.

For more information, please visit https://amberflo.io and get started for free.