Etherparty Whitepaper Draft

Lisa Cheng

March 31 2015

Smart Contracts driving new Service Oriented models for building the next generation of Software as a Service (SaaS) tools. Imagine Salesforce powered by a Smart Contract - with Etherparty your Smart Contracts can serve business processes from client lead to invoicing.

1 Etherparty design principles

Consistent Design - Naming Conventions for Etherparty will use the same name for parts of the contract that mean the same thing and different names for parts that are conceptually different. The naming standard applied for Etherparty (as per the Canonical Expression pattern) may be the only way to guarantee that contracts are consistent in how they express themselves. This includes choosing names that are human-readable and ideally, provide a business context.

1.1 Draft

Software as a service (SaaS)
delivered by Smart Contracts will have a major impact on the software industry. When SaaS first appeared in (snip) the software industry described it using words like 'revolution' and 'horizon.' It promised a future of software delivery that would change the way people build, sell, buy, and use software. The challenge was always how to develop Saas applications effectively and getting developers to start thinking about Saas deployment from the beginning, as opposed to the on-premise stack that was historically code intensive, a challenge to debug, and continues to be complex to improve upon.

Software as a Service - Defined "Software deployed as a hosted service and accessed over the Internet" - key here is the application code is residing, deployed, and accessed over the Internet - a vendor hosts all of the program logic and data, and provides end users with access to this data over the public Internet through a web-based user interface

Lines of Business services scalable solutions that solve business processes sold on a subscription basis

Consumer Oriented Services offered to the general public sometimes sold on a subscription basis but often provided at no cost, and supported by advertising

Changing the Business Model Software as a service has traditionally been offered by a central vendor who hosts the service, with Etherparty the Blockchain will be managing the message relays and transaction processing effectively removing the need for a central entity to host service oriented contracts. This new reality of Service oriented architecture will be driven by peer interactions who execute and manage their own service contracts.

This move from central Saas tools to peer created Saas will require a shift in thinking in three interrelated areas:

1. the business model

2. the application architecture

3. the operational structure ...

1. The Business Model is changing as it shifts the ownership of the Saas from provider to user 2. reallocating responsibility for the technology infrastructure and managementmaangement - from professional and hosted solution providers to third party development teams from Ethereum and Bitcoin, along with their decentralized network of consensus nodes. 3. As the cost of hosting and managing these services is moved to the Blockchain providers, the cost of providing services directly to users is reduced 4. Targeting the 'long tail' of smaller businesses, by reducing the minimum cost at which software can be sold

1.2 Who owns Smart Contracts?

It's important to note that software continues to be sold in the same way its been sold for decades, where the customer buys a license to use the software and installs it on the hardware that belongs to the customer or that is otherwise under the customer's control - with the vendor providing support as directed by the terms of the license or a support agreement. Legally - the customer is purchasing a right to use, but not own the actual code that sits on their hardware.

Software as a product model - where its sold via license and installed gives the feeling of ownership. Whereas the Saas model tells the customer they can pay for a subscription to a software running on someone else's servers and stops working if they stop paying.

Changing Software Delivery Traditional software costs take into account spending resources - both financial and human capital on: - software development - hardware including computers, servers, network components, and mobile devices that provide users with access to the software - professional services - the people and institutions that ensure the continued operation and availability of the system, including technical support staff, consultants, and vendor representatives

– Of these three, software is the most directly involved with information management - which is the ultimate goal of any IT organization. Professional services and hardware are typically considered a means to an end - in the sense that they make it possible for the software to produce the desired end result of effective information management. (In other words, any organization would gladly add software functionality without extra hardware if it could do so effectively but no organization would simply add hardware without an anticipated need to add software as well.)

With the traditional software delivery model of on-premise, the majority-majroity of the budget is typically spent on hardware and professional services unfortunately. On-premise business software spends most of their budget on licensedlincensed copies of shrink wrapped business software and customized line of business software. The hardware budget goes toward desktop and mobile computers for end users, servers, to host data and applications, and components to network them together. The professional services budget pays for a support staff to deploy and support software and hardware, as well as consultants and development resourcesresoureces to help design and build custom systems.

Saas environments currently hold critical applications and associated data on central servers at the vendor's location where it also supports the hwardware and software with a dedicated support staff. This relieves the customer organization from the responsibilityresponsbility for supporting the hosted software, and for purchasing and maintaining the server hardware for it. Moreover, applications delivered over the web or through smart clients place significantly less demand on a desktop computer than traditional locally installed applications, which enables the customer to extend the desktop technology lifecycle significantly. The end result is that a much larger percentage of the IT budget is available to spend on software -typically in the form of subscription fees to Saas providers.

Leveraging Economy of Scale Based on the economies of scale with Saas, allows Saas providers to scale their operational costs as the cost of each customer decreases as more customers are added. As this is happening, the provider will develop multi-tenancy as a core competencycomptenency leading to higher quality offerings at a lower cost. Therefore even accounting for the hardware and professional services costs incurred by Saas vendorsvenors customers can still obtain significantly greater pure software functionality for the same IT budget.

Selling to the Long Tail In 2004 Wired Magazine published an article by Chris Anderson that explained the concept of the Long Tail - that is the market space that traditional retailers are unable to reach as its not cost effective. The example used was how online retailers such as Amazon are uniquely positioned to fill a huge demand that traditional retailers cannot serve cost effectively. The underlying premise of the long tail market is a higher volume of small frequent sales, rather than appealing to the mass market best seller inventories.

Authentication In a decentralized authentication system, the tenant deploys a federation service that interfaces with the tenant's own user directory service. When an end user attempts to access the application, the federation service authenticates the user locally and issues a security token, which the SaaS provider's authentication system accepts and allows the user to access the application