



Open Source Compared with Commercial SIP Stack: Case Study

Introduction

Developers of SIP-based products have several sources of SIP stacks to build into their products. It is understandable that many such developers would consider “free” open source SIP stacks as one way to go, but the initial cost of a SIP stack is only one factor in considering the overall cost, complexity and risk of open source compared with a commercial protocol stack.

This document provides a case study of the pros and cons of a commercial SIP stack, CompactSIP, compared with an open source stack, Sofia-SIP.

When considering the costs related to SIP over a product’s lifetime, there are several key factors: cost of learning the SIP stack, ease-of-use of the API, and maintaining and enhancing the product. A short video <http://www.telesoft-intl.com/video/index.html> discusses key issues in considering open source vs. commercial solutions on a general level.

CompactSIP

CompactSIP is a fully-featured, well-documented, and proven SIP source code stack from TeleSoft International. CompactSIP is notable for its very small memory footprint and ease of portation and integration to a variety of applications running under a range of operating systems/environments. With customers in over 20 countries on six continents, TeleSoft International, Inc., (www.telesoft-intl.com) has been a supplier of quality protocol stacks and superior technical support since 1992.

Sofia-SIP

Sofia-SIP is an open source SIP stack with a subset of the features of CompactSIP. It is a project developed and supported by a couple of engineers in Nokia and other contributors on a volunteer basis.

CompactSIP vs. Sofia-SIP: Differences at a glance

Attribute	CompactSIP	Sofia-SIP
Reflects experience building & supporting protocol stacks, with resulting quality of architecture & code design (reflected in footprint, ease of development)	High	Low
Modular design enables optimization to application	Yes	No
Memory footprint	75-150 KB	600 KB+
Complexity based on number of API calls	14	> 100
Other SIP-related application complexity	Low	High
Ease of using pre-developed apps as templates	High	Low
Portable design with support for multiple OSes (11+)	High	Low
Quality of documentation and code comments	High	Medium
Comments as % of example app lines of code	10%	5%
Tech support responsiveness & quality	High	Uncertain
Cost of stack and example app source code	Medium	Low
Learning curve & initial development cost	Low	High

Open Source Compared with Commercial SIP Stack

Maintenance & enhancement cost	Low	High
Extensive interop testing	High	Uncertain
Development team	Focused	Diffuse
Schedule risk (complexity increases risk)	Low	High

Experience Developing and Supporting Protocol Stacks

In developing CompactSIP as a coherent, robust, easy-to-use technology building block, TeleSoft has leveraged its many years of experience in developing communications protocols, and very importantly, supporting customers integrating TeleSoft stacks into their products.

Successful commercial stack providers make their living by how well designed and documented their products are; they have a built-in motivation to continually offer better products. Open source, in contrast, is a volunteer effort, not for profit, even when a big company like Nokia is involved.

Communications building block development is very specialized and the product is different from most software because it is distributed in source rather than binary form. Few organizations have the necessary experience developing such specialized source code products. TeleSoft brings to its SIP stack architecture and code design decades of experience helping hundreds of companies worldwide develop their communications products.

By contrast, software developed by a volunteer company or ad-hoc committee just doesn't have the built-in mechanisms for cleanliness and coherence of design that commercial products do. While Nokia successfully sells phones, Sofia-SIP support is a revenue loser for them, and supporting others in using protocol stacks is not their business.

The disconnection of developers and support from revenue generation creates a major disadvantage for Sofia-SIP users: market forces do not operate to weed out overly complex and confusing code. To the contrary, open source by nature has built-in impediments to simplicity, conceptual integrity and consistency.

Open source that is overly complex can survive because of the initial price, and the unwary developer will find out the hard way, due to difficulty of development and inevitable schedule slips. Only then, when it is too late, do many developers see that the pitfalls of open source are real, and that the cost can actually be greater over the initial development period, and even greater over the lifetime of a product. The adage "you get what you pay for" really applies here.

Memory Footprint

The Sofia-SIP executable code memory footprint is greater than four times the size of CompactSIP: Sofia-SIP size is greater than 600KB (Text segment) vs. CompactSIP size of 75-150KB (Text segment). This difference significantly impacts performance and power consumption in many applications. (Reference: <http://osdir.com/ml/telephony.sofia-sip.devel/2007-01/msg00076.html>)

Complexity Based on Number of API Function Calls

The CompactSIP API has *14 API function calls*. To perform the same functions, Sofia-SIP has well over one hundred API calls (Sofia-SIP is so complex that it is hard to determine exactly how many calls there are!).

Other SIP-related Application Complexity

Using Sofia-SIP to develop, maintain and enhance applications is significantly more complex compared with CompactSIP based on factors beyond the number of API calls. Some examples:

Thread Management

Sofia-SIP specifies, and is entangled with, thread/task management. This limits portability and unnecessarily complicates/restricts applications, because Sofia-SIP developers have to

Open Source Compared with Commercial SIP Stack

follow a particular thread management model. In contrast, the CompactSIP API is a simple function library that makes no assumptions about how it is called and imposes no thread/task structure. CompactSIP is called as a library independent of threads, tasks, processes, polling, or other OS scheduling mechanism. OS scheduling dependencies should not be part of a well-architected SIP stack.

Protocol Abstraction Complexity

Each stack requires an abstraction layer to be able to provide application access to SIP services and message elements. CompactSIP has a simple mechanism of a few API calls and a few dozen elements for a simple, uniform SIP message access method. Sofia-SIP uses a much more complex tag-based interface with an order of magnitude more elements to perform the same functions.

Ease of using pre-developed applications as templates

CompactSIP provides a choice of key applications with a high-degree of fit-and-finish, portability and quality documentation:

- *TsSmartPhone* SDK for mobile, wired or soft SmartPhones, highly portable, and requiring an absolute minimum of application functionality to implement a fully featured VoIP phone.
- *TeleSoftPhone* softphone reference design, highly portable, and requiring an absolute minimum of application functionality to implement a fully featured VoIP phone.
- *TsGATE* SIP-PSTN gateway application leveraging CompactSIP with TeleSoft's extensive, proven TsLink3 protocol stack for ISDN, T1 RBS and E1 R2 for major world markets.
- *TsCONNECT* IP-PBX Voice to PSTN Connectivity SDK enables suppliers to easily add host-based PSTN connectivity to their IP-PBX using a SIP API.

By contrast, the complexity of the Sofia-SIP API makes it difficult to analyze and use the poorly documented example apps, which require more work for the developer compared with CompactSIP.

Quality of Documentation and Code Comments

CompactSIP documentation is succinct, top-down, easy to get started with. Sofia-SIP documentation is overwhelming, chaotic and hard to comprehend.

CompactSIP code comments are extensive and descriptive. Comparison of several thousand lines of code in the app template code available with CompactSIP (cpapp.c) vs. Sofia-SIP (ssip.c) shows that CompactSIP has about 10% lines of code with comments while Sofia-SIP has about 5%.

Tech Support Responsiveness & Quality

Similar to most open source software, a look at the July, 2008, sofia-sip-develop mailing lists reveals a disquieting set of problems unresolved a month later, including many that would show up in most applications. Without a direct connection to the stack supplier, the user has no leverage, no priority and no certainty of receiving any timely, authoritative tech support.

By contrast, TeleSoft typically responds within 24 hours and usually has each issue resolved within days. TeleSoft tech support is provided by senior software engineers who develop the code and is provided promptly by email, by phone and by interactive web-based screen sharing sessions.

SIP-related Learning Curve & Product Development Cost

The initial cost of open source software like Sofia-SIP is lower than a commercial alternative. But that expense is typically not the biggest part of the development cost. Increasing development time by a factor of 2 to 3 can more than offset a difference in cost of the starting point software, with the inevitable increased schedule risk of missing a market window.

Lifetime Product Costs: Maintenance and Enhancement Costs

Maintenance and enhancement costs over the lifetime of a product are frequently significant, sometimes larger than initial development. When engineers are tasked to update or fix software that no one has looked at in months or years, it is critical that they inherit an easy-to-understand, well-documented and responsively supported protocol stack.