Building an Enterprise API & Developer Community with Atmosphere

atmosphere™ SOA | software™

soa.com/atmosphere

TABLE OF CONTENTS

Introduction 3

Creating a successful API 3

Building the right API 3

Publishing and Sharing your API 4

Defining and Documenting your API 4

Ensuring that Developers can Find your API. 5

Managing Connections Between Apps and APIs 5

Protecting your API 6

Security 7

Quota Policy 7

Socializing your API 8

Invitations 8

Social Forums 8

Following & User Dashboards 8

Supporting your App Developers 9

Managing APIs for B2B Integration 9

Sharing Developer Communities 10

About SOA Software Atmosphere™ 11

App-Centric APIs 11

Policy-Driven 12

APIs Your Way 12

Start From Scratch, Use What You Have, Or Both . . 12

How does Atmosphere Work? 13

About SOA Software 14

INTRODUCTION

The API Economy is here. Enterprises are making business applications available through APIs to drive business growth and expose new opportunities. The business landscape is being reshaped as dramatically as it was during the rush to create an Internet presence with a web site in the late 90s. APIs are becoming the primary way that businesses interact with their customers, reach new markets, and provide the global app development community with the tools to deliver innovative new business capabilities to customers.

An API is really a product, and has to be thought of with the same rigor as any product. It is important that you identify the right business capability to expose as an API, the API is well designed and built, runs correctly, is well supported, and is promoted effectively. This paper walks through the steps you need to go through in order to help ensure your API program succeeds.

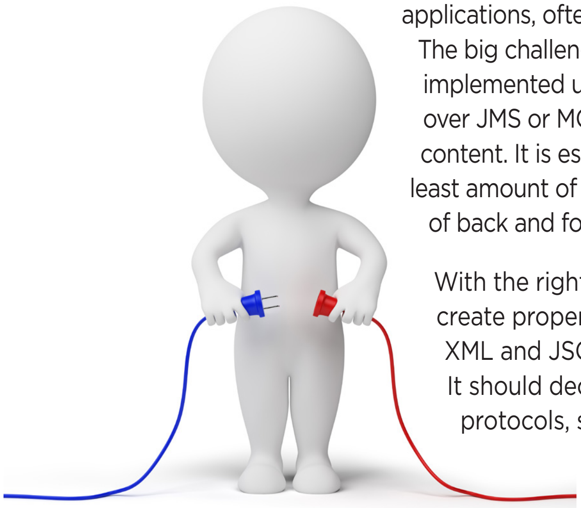
CREATING A SUCCESSFUL API

Whether you're just starting down the path with APIs, or you already have a great API in production, you need a solution that deals with all the things you don't want to have to deal with. You should be focused on the business value and functionality of your API, rather than worrying about making sure your API is running right. In other words, if it has nothing to do with your core business functionality, leave the heavy lifting to an API and Management and Collaboration solution.

Leave the heavy lifting to an API and Management and Collaboration solution.

Building the right API

APIs don't typically exist in a nice little self-contained package installed in your enterprise DMZ. In most cases APIs will be an externally facing interface to one or more internal applications, often using existing service interfaces to create the API. The big challenge is to take an existing service interface that is likely implemented using SOAP with WS-Security, or Plain-Old-XML (POX) over JMS or MQ, and turn it into a RESTful API with XML and/or JSON content. It is especially important with mobile applications to deliver the least amount of data possible to get the job done, with the least amount of back and forth dialog.



With the right API Management solution you can quickly and easily create properly defined and structured RESTful APIs supporting XML and JSON that are built on top of existing internal services. It should declaratively mediate between content types, transport protocols, security models, message exchange patterns, standards and more. This will allow you to get your API up and running quickly.

Publishing and Sharing your API

So you've just built the best API in the history of APIs, and you're going to have tons of businesses using it, or tens or even hundreds of thousands of Apps with millions of users, and the new business will keep pouring in generating vast profits and turning you into the next billionaire, or at least netting you a pay rise, or some good kudos inside your company. The only problem is that you really don't know how to tell people about it, and when you do find someone who's interested, you have to be able to feed them all the information they need to use it without getting overwhelmed trying to support them all.

This is where you need a social API management and collaboration platform, regardless of whether you are trying to broadcast your Open API to the whole wide world, or to expose it to groups of developers inside your own business.

You want to market your API or App inside or outside your enterprise with a powerful search driven catalog offering rich social features like ratings, reviews and 'following'. You need to publish your API into a catalog with appropriate descriptions and tags to make it easy for potential consumers to find it through search and browse, including external (public or enterprise) search engine integration and optimization.

Of course, once someone has found your API, they need to decide if it is the right API for them to use. This is where definition and documentation capabilities will come into play, and where discussion forums and user-following capabilities will allow users to ask questions, or see how active the community around your API is.

Defining and Documenting your API

People need to be able to use your API without burdening you with a ton of questions and complaints. You need to ensure that you define it properly, document it well, and publish the definition and documentation effectively.

A good API management and collaboration solution will provide a robust set of tools for API definition, documentation, and content management, as well as policy and lifecycle management. This will allow you to make sure your APIs are correctly structured and well documented. The solution should also integrate well with internal design and development processes to help make sure that the internal services and applications that support your API are up to the task.

Ensuring that Developers can Find your API

Good stuff, you've defined your API properly and have published it with well-structured documentation. It's in the catalog with a nice description, an appropriate name and set of icons, and you've even convinced a few friends to post nice encouraging reviews and ratings for it. Now it's time to hope that potential users can find it.

Your API management and collaboration solution should provide a very powerful built-in system that creates an internal search index from all the content about the APIs and Apps that it is aware of, complete with sufficient permission information so that it knows which users it can show search results for particular APIs and Apps to.

Hopefully the phrase "all the content" caught your eye, because that's one of the more interesting things here. The content about your API or App doesn't just include its name and any description information you typed in, it also includes all the documentation, and most importantly all the discussion topics and comments posted about it. This means that if someone asks a question in your API's forum about how it handles a particular type of widget, then when someone searches for anything to do with that widget, they should find your API as well as the question (and hopefully the response). Think about the use-case where you are writing a piece of code and you run into a problem and use Google to search for the specific error code or exception text. Because a good API management and collaboration solution will index all this information, these types of searches should work in real-time.

Another important capability for your solution's search system is its ability to publish and maintain its index in enterprise or public search engines. This will allow your developers to find information about your API regardless of how they look for it, or which tools they are using.

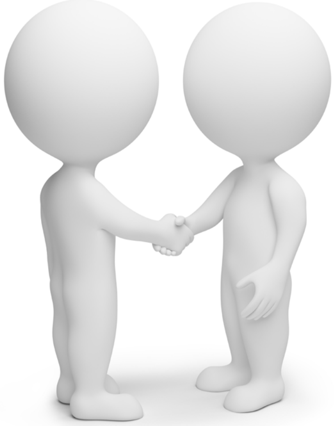
People need to be able to use your API without burdening you with a ton of questions and complaints.

Managing Connections Between Apps and APIs

You're making real progress with your API now. It's well defined and documented, you've published it into a dynamic catalog where users are able to find it, rate it, review it and discuss it. What more could you possibly want?

How about allowing Apps to actually use your API, and controlling the way they use it? This is where your API management and collaboration solutions come together. The combined solution will allow developers to define their Apps – in much the same way that you defined your API. In fact, the App definition should have all the same features with documentation, ratings, reviews, discussion boards and the whole nine yards, as your API definition. It should also allow the developer to create a team of peers that can work together on the App. This may all seem a bit fluffy and useless, until we get to the real purpose of the App definition.

Your API management and collaboration solution should provision API access individually to each App. It should create an identifier for the App, allowing the App developer to upload a CSR (certificate signing request) for a locally generated public key. The App should then use this App-id (optionally signed) to authenticate itself to the API Management solution. The App developer should use the Developer Community Management solution to find the API(s) they want their App to use, and initiate a simple approval workflow to grant their App access to the APIs. Depending on how the API is configured, they should also be able to request different throughput levels based on pre-defined quota policies. Once the workflow process completes and their App's access to the API(s) is approved, the App will be able to begin sending requests to the API.



And that's not all. You need the ability for an API to expose sandbox (test) and production endpoints, with a multi-step approval process for granting Apps access to each of the endpoints. This may sound complex, but in most circumstances, sandbox access should be automatically approved, and getting approval for production access should be a simple process involving a single button click for the App developer, and a simple response to a newsfeed item and/or notification from the API administrator.

One of the interesting side-effects of this provisioning process is that your solution should be able to monitor the API from the perspective of each of the Apps. The API administrator will be able to see all of the monitoring data and can choose to see a chart showing which Apps are consuming their API the most, they should be able to filter the monitoring data by App to provide support to a particular App owner, or use this data to identify poorly written or malicious Apps that are causing problems for their API, so they could choose to disable or throttle them appropriately. Similarly, the App owner would see the API from the perspective of their App only. They would only see their traffic and performance information, and the usage data and messages their App generates.

A well-designed API Management solution will keep things as simple as possible.

Protecting your API

It's all very well to publish a great API, but there are a couple of things you have to be wary of:

- First off, does your API expose sensitive information or even enable business transactions? For example, it would be a really bad idea to overlook security when you deliver an API that exposes private information about your customers, or allows people to move money from one account to another.
- Regardless of whether your API needs sophisticated security or not, it will still be vulnerable to abuse, intentional or otherwise. You need to be able control the amount of traffic each Application is able to send to your API to ensure that one App can't dominate the API, preventing other Apps from using it; and to protect your internal systems from the damage that could result from the API overloading them with traffic.

Security

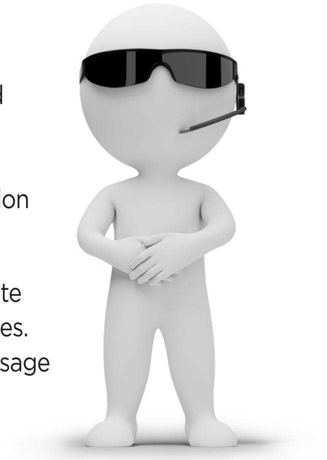
A good API Management solution will provide a rich array of security capabilities for all types of APIs, including the ability to mediate between security models, standards and technologies to ensure seamless interoperability between externally facing APIs and the internal services that support them. These security capabilities should include:

- **Authentication** – at a minimum your solution must be able to authenticate the Apps that are consuming your APIs. It should support a wide array of token types from simple http headers (unsigned or signed), through x.509 certificates and highly advanced WS-Security headers.

- **Token Exchange** – the solution should be able to take the credentials provided by the App and use them to generate a whole new token as needed by your internal API. For example, it could authenticate an App using a signed http header, and generate a signed WS-Security SOAP transport, or a SAML assertion for use by your internal service.

- **Cryptography** – the solution should provide a built-in PKI system and Certificate Authority for generating and distributing public/private keypairs and certificates. It should provide policies for specifying a wide range of different types of message signature and encryption supporting all common standards.

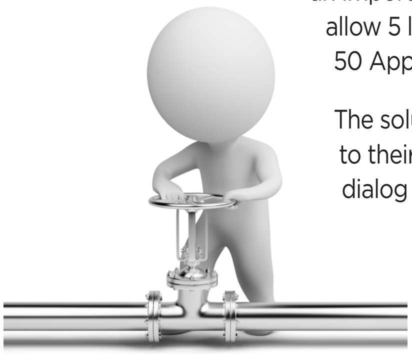
- **Federation** – the solution should allow an App to use a single credential to consume many APIs, even if the APIs are offered by different companies with their own API Management platforms. For more on this see the section below on API provider federation.



All these capabilities and the huge range of supported standards and configuration options could make for a very complicated product, but a well-designed API Management solution will keep things as simple as possible, providing canned configurations for common scenarios, with a comprehensive set of wizards for more advanced use-cases.

Quota Policy

An API management and collaboration solution should provide a powerful quota policy system allowing API administrators to provision capacity to Apps as needed. This means that an administrator who knows how much traffic the API can handle could apportion this capacity over the Apps that are consuming the API appropriately. For example if an API can handle no more than 200 transactions per second (tps) the administrator may choose to allow an important customer facing application to consume 100tps, and then can allow 5 less important Apps to consume only 10tps each, and the remaining 50 Apps are only allocated 1tps.



The solution should allow App developers to negotiate the quota allocated to their App during the access provisioning process, and can facilitate a dialog between the App developer and the API administrator if the App needs more or less capacity at any point in the future. It should then enforce the quota policies at runtime, denying or queuing up any requests that exceed an App's allotment.

Socializing your API

Assuming you're following along with the various steps above, then by this point you now have a rock solid API that is well structured and documented, published in a catalog with great search facilities, and is protected from all kinds of bad things. Developers are starting to find it through the traditional search and browse models, but that's not good enough in today's social-media driven world. You need your API (or App) to go viral. Bring on the social world.

An API management and collaboration solution will provide the social platform for API Management and App development. It will bring together API providers and App developers in an online social community. Whether this community exists inside your enterprise, as a private external community, or part of a broad public developer community is entirely up to you. The solutions social features should include:

Invitations

Any user, developer, business administrator, or API provider should be able to invite other users to the party. Invitations should be able to be general, simply inviting a user to check out the specific Atmosphere instance, or they could invite a user to follow an API, App, or Group, or join the development team for an App. In this way you can immediately create your own communities of interest around your APIs and Apps. By encouraging the people you invite to invite their contacts too, you can build a huge social community connected to your resources.

You need your API or App to go viral. Bring on the social world.

Social Forums

Every API, App, User, Business or Group inside your Developer Community solution should have its own social forum – think Facebook Wall. This is a place where users can go to post comments or ideas, ask questions, or even raise support tickets. Depending on a user's role they should get different views of the forums. The administrator of an API, for example, should see access requests and unanswered trouble tickets that would be invisible to regular users. The administrators would use the forum as a central place to manage the workflow of access requests and trouble tickets, and to make sure that any questions and comments are being answered. This is a key concept - it's not necessarily up to the administrators to answer questions and even tickets themselves - as the community around a resource matures, it will likely become more and more self-sufficient, with many questions and comments being answered by other users.

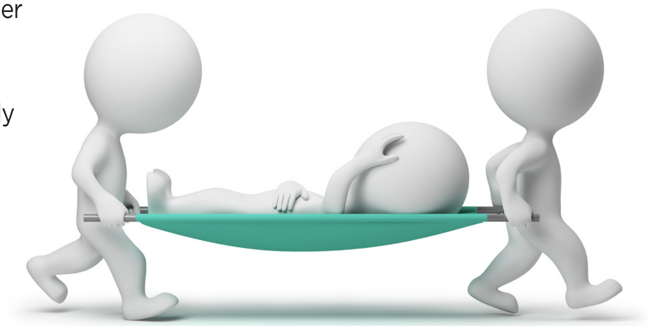
Following & User Dashboards

A standard old forum is a bit passé these days, so your API management and collaboration solution needs to be far more social. It should implement a Twitter-like following concept allowing users to follow APIs, Apps, Businesses, Groups, or even other users. Each user should have their own dashboard – like a Facebook news feed – that aggregates all the discussion items and comments from all of the resources the user is following. This would give the user a single centralized place to keep track of what's going on with everything they are interested in.

Supporting your App Developers

One of the bigger challenges you'll face with your API is the potential of becoming a victim of your own success. What's going to happen when you drive this massive social adoption of your API? How on earth are you going to manage the volume of support requests and questions you're sure to get?

Again, your API management and collaboration solution should have the answers. The first thing, of course, is to make sure that your API is well structured and well documented, as we discussed earlier. You can make developers' lives even easier by providing them with SDKs and code samples, and then be able to publish and promote these with your API. But where the real beauty kicks in is when your developer community becomes self-sufficient and users start helping each other out by answering questions and even responding to trouble tickets. Because everything should be so efficiently indexed, it should be really easy for users to search for errors, issues or questions, and get immediate help and answers from questions and issues that have already been resolved. Sure, you'll have to put some effort into getting your community moving, but the time and energy you dedicate early on will pay enormous dividends later. As my mother used to say, "A stitch in time saves nine."



Managing APIs for B2B Integration

Of course not every API will be destined to be truly "Open" - designed to target a broad community of App developers sharing ideas and applications in an open forum. In fact, we expect to see far more private or semi-private APIs that a business uses to share specific functions or data with one (private) or more (semi-private) partners.

A good API management and collaboration solution should enable this by introducing the concepts of API and App privacy settings, enabling a number of key scenarios.

In one of the most common scenarios you may have an API that you wish to use to enable integration with several partners, but you don't want any of the partners to know anything about each other, or even to know that there are other partners using the API. Your API management and collaboration solution should allow you to define a private API and invite users to follow and connect to it – the invitations will be required, because the API would not be published in the search indexes and would therefore not be visible to any user who has not been specifically invited. In this mode, user posts, discussions and comments would be visible only to that user and to the API administrators.

Another common requirement for private APIs is the need to create an API that is visible and consumable by a defined (invited) group of users. Any member of the group would be able to see posts and comments from other members, but the API and all its content would be hidden from anyone outside that group. This is particularly useful as a tool for publishing a version of an API for use by a beta test group, in advance of launching the API to the public.

Sharing Developer Communities

All these capabilities sound great, but at the end of the day, the success of your API initiative depends very much on your ability to get your API in the hands of as many developers as possible. So how can you go about building your own developer community? Maybe a better question would be, why should you go about building your own developer community when there are already a bunch of communities out there? More to the point, what about your partners or other like-minded companies with their own APIs that a developer building an App that uses your API might also use?

Your API management and collaboration solution should support the concept of API provider federation to bring together communities giving developers access (with approval) to any API from any provider using a single App ID, and allowing providers to leverage the network effect of all the connected communities to find the broadest possible reach for their API. All this should be done with federated trust and permission models that allow API providers to opt in or out at granular levels, and to choose the partners with whom they want to federate.

An API management and collaboration solution that is offered as a SaaS platform as well as via on-premise or hybrid models would provide a ready-made community for its customers to connect to, whether they choose to become a user of the SaaS platform, or install the product in their own datacenters. Customers should be able to choose how much or little sharing they do with the broader community, depending largely on how 'Open' they view their APIs.

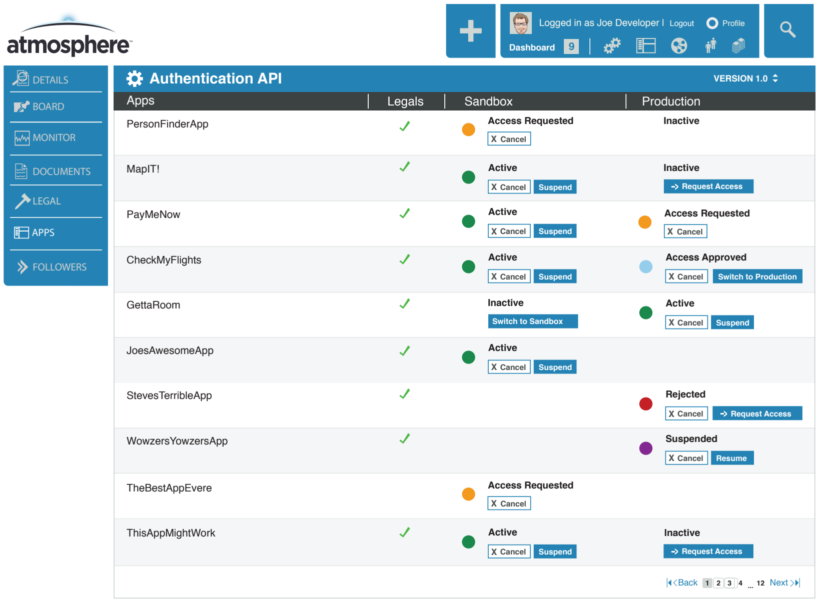
ABOUT ATMOSPHERE

The SOA Software Atmosphere™ product family was created to help drive the API Economy by meeting the needs of collaborating around and managing APIs in a complex environment. It provides a secure, robust platform that companies can use to share their APIs with the developer community of their choice. Atmosphere manages, monitors, and secures companies' APIs ensuring that they deliver the level of service customers and partners require; the security of corporate and customer information and assets; and the integrity of the corporate brand.

Atmosphere is an API management and collaboration product family that brings together API Providers and Application Developers in a community which should feel familiar to anyone who spends time on popular social networks.

It provides a connecting point where your APIs can be published, promoted, and managed in a secure, scalable environment. You can manage your own developer community, or just plug into communities that already exist. Your APIs can be discovered and harnessed by creative people who blend it with complementary APIs from your partners. These developers can then work together, share ideas, and support each other in new and productive ways.

App-centric visibility into API usage.



Apps	Legals	Sandbox	Production
PersonFinderApp	✓	● Access Requested X Cancel	Inactive
MapIT!	✓	● Active X Cancel Suspend	Inactive → Request Access
PayMeNow	✓	● Active X Cancel Suspend	● Access Requested X Cancel
CheckMyFlights	✓	● Active X Cancel Suspend	● Access Approved X Cancel Switch to Production
GettaRoom	✓	Inactive Switch to Sandbox	● Active X Cancel Suspend
JoesAwesomeApp	✓	● Active X Cancel Suspend	
StevesTerribleApp	✓		● Rejected X Cancel → Request Access
WowzersYowzersApp	✓		● Suspended X Cancel Resume
TheBestAppEvere		● Access Requested X Cancel	
ThisAppMightWork	✓	● Active X Cancel Suspend	Inactive → Request Access

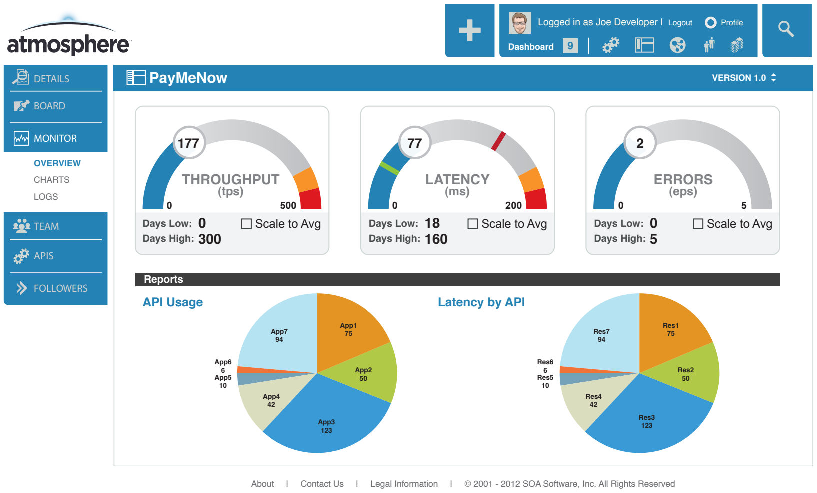
There is a lot of social goodness here, but we haven't lost sight of the fact that the most important thing is to meet the needs of the day-to-day management of your APIs and the applications that use them.

Atmosphere stands on the shoulders of a rich heritage - SOA Software has been around the block a few times. We work with many of the Fortune 1000, meeting the needs of demanding production environments for SOA governance.

App-Centric APIs

Atmosphere provides sophisticated yet simple security capabilities that allow API providers to control and monitor app's usage of their APIs.

Heads up display to let you know what your API is up to.



Days Low: 0 ☐ Scale to Avg
Days High: 300

Days Low: 18 ☐ Scale to Avg
Days High: 160

Days Low: 0 ☐ Scale to Avg
Days High: 5

Policy-Driven

Atmosphere defines all security and quality-of-service through centrally managed policies, ensuring consistency between multiple API implementations and versions.

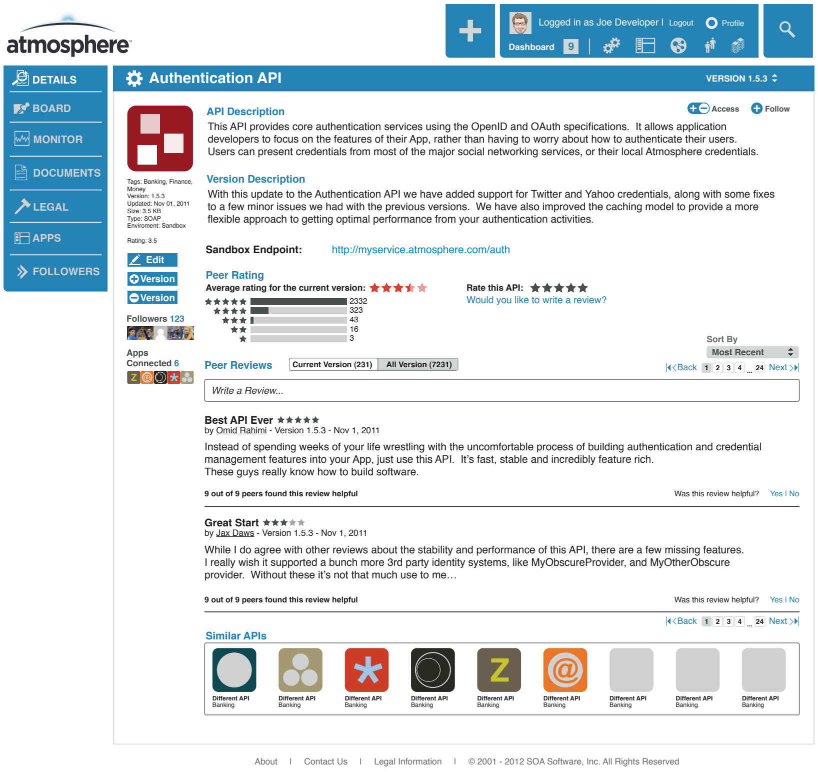
APIs Your Way

Atmosphere allows you to define and manage your APIs using a wide range of messaging types and formats including REST/XML, REST/JSON and SOAP. You can create an API with multiple interfaces using different standards and different security mechanisms with no extra effort.

Start From Scratch, Use What You Have, Or Both

In addition to exposing new capabilities, you can leverage Atmosphere to harness existing investments in web services and enterprise SOA, with new formats and policies specific to an outside developer group.

Atmosphere provides a lot of information about an API, including how it is being used and how popular it is.



Authentication API VERSION 1.5.3

Tags: Banking, Finance, Money
Version: 1.5.3
Updated: Nov 01, 2011
Size: 3.5 KB
Type: SOAP
Enviroment: Sandbox

Rating: 3.5

Followers 123

Apps Connected 6

API Description

This API provides core authentication services using the OpenID and OAuth specifications. It allows application developers to focus on the features of their App, rather than having to worry about how to authenticate their users. Users can present credentials from most of the major social networking services, or their local Atmosphere credentials.

Version Description

With this update to the Authentication API we have added support for Twitter and Yahoo credentials, along with some fixes to a few minor issues we had with the previous versions. We have also improved the caching model to provide a more flexible approach to getting optimal performance from your authentication activities.

Sandbox Endpoint: http://myservice.atmosphere.com/auth

Peer Rating

Average rating for the current version: ★★★★☆

★★★★★	2332
★★★★	323
★★★	43
★★	16
★	3

Rate this API: ★★★★★
Would you like to write a review?

Sort By
Most Recent

Peer Reviews Current Version (231) | All Version (7231)

Back 1 2 3 4 ... 24 Next

Write a Review...

Best API Ever ★★★★★
by Omid Rahimi - Version 1.5.3 - Nov 1, 2011

Instead of spending weeks of your life wrestling with the uncomfortable process of building authentication and credential management features into your App, just use this API. It's fast, stable and incredibly feature rich.
These guys really know how to build software.

9 out of 9 peers found this review helpful Was this review helpful? Yes I No

Great Start ★★★☆☆
by Jax Daws - Version 1.5.3 - Nov 1, 2011

While I do agree with other reviews about the stability and performance of this API, there are a few missing features.
I really wish it supported a bunch more 3rd party identity systems, like MyObscureProvider, and MyOtherObscure provider. Without these it's not that much use to me…

9 out of 9 peers found this review helpful Was this review helpful? Yes I No

Back 1 2 3 4 ... 24 Next

Similar APIs

Different API Banking	**Different API** Banking	**Different API** Banking	**Different API** Banking	**Different API** Banking	**Different API** Banking	**Different API** Banking	**Different API** Banking	**Different API** Banking

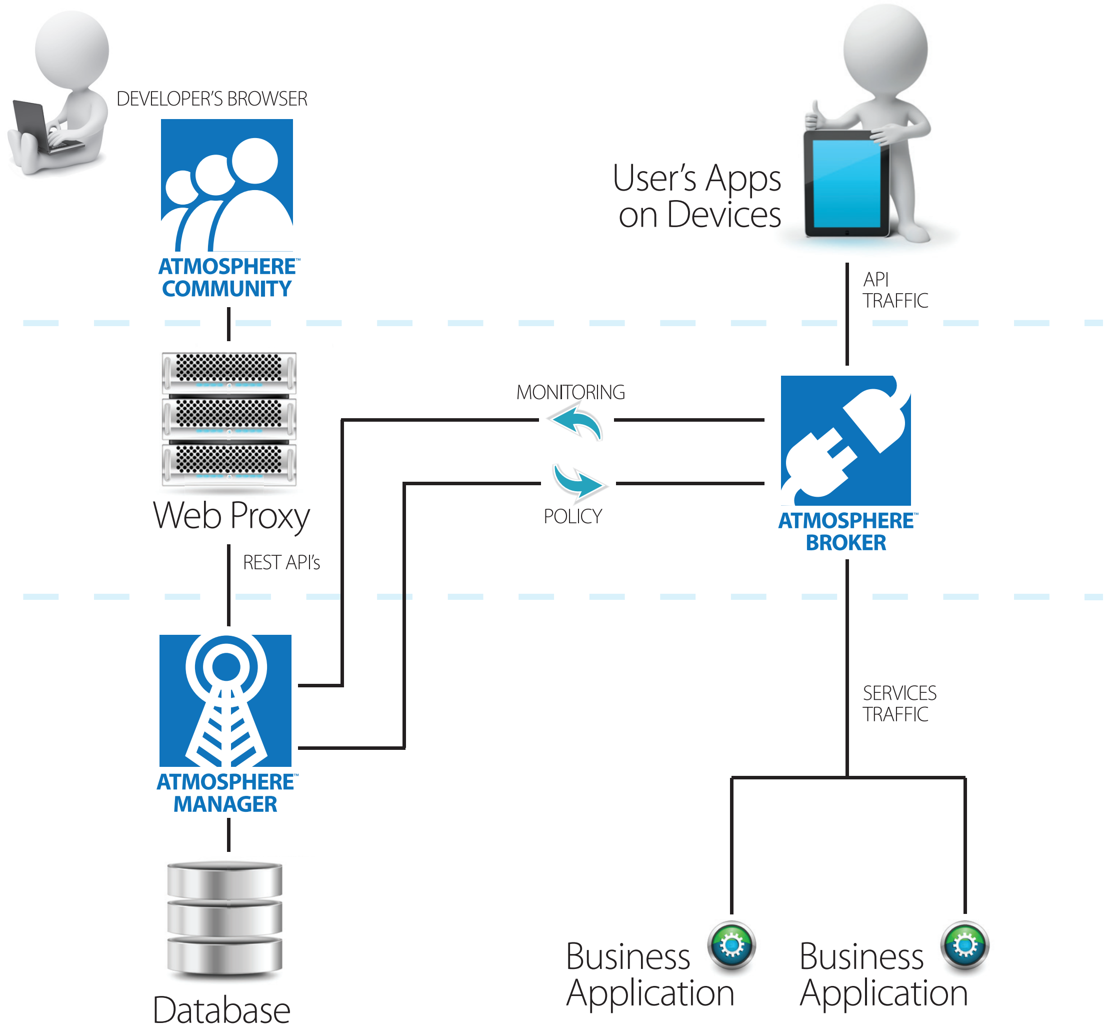
About I Contact Us I Legal Information I © 2001 - 2012 SOA Software, Inc. All Rights Reserved

How does Atmosphere Work?

The Atmosphere product family is built on three main components:

- **Atmosphere Manager** – provides the services and APIs that support Atmosphere Broker and the Atmosphere Community UI
- **Atmosphere Broker** – an API proxy server providing security, monitoring, mediation and other runtime capabilities
- **Atmosphere Community** – a sophisticated JavaScript client that runs in a web browser to extend API Management capabilities to a social developer community

The products are available as on-premise software, as Software-as-a-Service, or as a hybrid of both (typically leveraging the SaaS platform for the developer community and keeping API endpoint proxies inside the enterprise).



ABOUT SOA SOFTWARE

SOA Software is a leading provider of unified SOA governance, cloud and enterprise API Management products that enable organizations to plan, build, and run enterprise services and open APIs. The world's largest companies including Bank of America, Pfizer, and Verizon use SOA Software solutions to transform their business. For more information, please visit http://www.soa.com.

SOA Software, Atmosphere, atmos.phe.re, Policy Manager, Portfolio Manager, Repository Manager, Service Manager, and SOLA are trademarks of SOA Software, Inc. All other product and company names herein may be trademarks and/or registered trademarks of their registered owners.

SOA™ software

12100 Wilshire Blvd, Suite 1800
Los Angeles, CA 90025
866-SOA-9876
www.soa.com
info@soa.com

Disclaimer: The information provided in this document is provided "AS IS" WITHOUT ANY WARRANTIES OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF INTELLECTUAL PROPERTY. SOA Software may make changes to this document at any time without notice. All comparisons, functionalities and measures as related to similar products and services offered by other vendors are based on SOA Software's internal assessment and/or publicly available information of SOA Software and other vendor product features, unless otherwise specifically stated. Reliance by you on these assessments / comparative assessments are to be made solely on your own discretion and at your own risk. The content of this document may be out of date, and SOA Software makes no commitment to update this content. This document may refer to products, programs or services that are not available in your country. Consult your local SOA Software business contact for information regarding the products, programs and services that may be available to you. Applicable law may not allow the exclusion of implied warranties, so the above exclusion may not apply to you.