

**پروژه‌ي طراحی، پياده‌سازي و اجرای**

**زیرساخت تبادل اطلاعات سلامت**

**عنوان سند: توصیف فنی زیرساخت HIX و شیوه‌ی اتصال به آن**

**شناسه‌ي سند:** HIX.TEC.SPC.V.0.3.4.1

**نگارش:** V.0.3.4.1

**تاريخ تهیه:** 08/06/94

**شرح سند:**

اين سند شامل توصيف زیرساخت HIX به منظور تسهیل انجام تعاملات الکترونیکی بین سامانه‌های نرم‌افزاری در وزارت بهداشت است.

**فهرست مطالب**

[1- مقدمه 4](#_Toc429956545)

[2- انتقال اطلاعات در زیرساخت HIX 5](#_Toc429956546)

[1-2- مدل مفهومی 5](#_Toc429956547)

[2-2- توصیف لایه انتقال 7](#_Toc429956548)

[1-2-2- متعاملین 7](#_Toc429956549)

[2-2-2- نقاط انتهایی ارتباط 8](#_Toc429956550)

[3-2-2- زیرساخت 14](#_Toc429956551)

[3- تحویل مطمئن پیام و امنیت در زیرساخت HIX 16](#_Toc429956552)

[1-3- تحویل مطئمن 16](#_Toc429956553)

[2-3- امنیت 20](#_Toc429956554)

[1-2-3- امنیت دستیابی به کانال دسترسی 20](#_Toc429956555)

[2-2-3- امنیت دستیابی به سرویس‌های مقصد 21](#_Toc429956556)

[4- فراخوانی کانال دسترسی 22](#_Toc429956557)

[1-4- فراخوانی کانال دسترسی به‌صورت همگام 22](#_Toc429956558)

[1-1-4- ارسال پیام به وب سرویس کانال دسترسی 22](#_Toc429956559)

[2-1-4- بازگشت پاسخ از وب سرویس کانال دسترسی 23](#_Toc429956560)

[2-4- فراخوانی کانال دسترسی بصورت نا همگام 24](#_Toc429956561)

[1-2-4- ارسال پیام به یک مقصد 25](#_Toc429956562)

[2-2-4- ارسال پیام به وب سرویس کانال دسترسی 25](#_Toc429956563)

[1-2-2-4- بازگشت پاسخ از وب سرویس کانال دسترسی 26](#_Toc429956564)

[3-2-4- تاییدیه رسید پیام به مقصد در ارسال ناهمگام 27](#_Toc429956565)

[4-2-4- خطاهای زمانی در ارسال ناهمگام 28](#_Toc429956566)

[3-4- مدیریت خطا به صورت عمومی 29](#_Toc429956567)

[1-3-4- بروز خطا در فراخوانی کانال دسترسی 29](#_Toc429956568)

[4-4- امنیت در فراخوانی سرویس 31](#_Toc429956569)

[1-4-4- ثبت نام هویت سازمان و نرم‌افزار مبداء فراخوانی کننده 32](#_Toc429956570)

[2-4-4- صدور شناسه و سرفصل امنیتی برای متعاملین سوئيچ 33](#_Toc429956571)

[5-4- نمونه کد فراخواني کانال دسترسی زیرساخت HIX به صورت همگام 34](#_Toc429956572)

[1-5-4- نمونه کد در .NET 34](#_Toc429956573)

[2-5-4- نمونه کد درJAVA 35](#_Toc429956574)

[6-4- سرویس‌های زیر ساخت 36](#_Toc429956575)

[1-6-4- دریافت لیست تفاهم‌نامه‌های مربوط به سازمان 36](#_Toc429956576)

[2-6-4- آگاهی از وضعیت سرویس در سازمان مقصد 39](#_Toc429956577)

[5- پیوست‌ها 42](#_Toc429956578)

[1-5- پیوست 1: توصیف مدل ارتباطی همگام 42](#_Toc429956579)

[2-5- پیوست 2: توصیف مدل ارتباطی ناهمگام 42](#_Toc429956580)

1. مقدمه

در سالیان اخیر روند الکترونیکی شدن تبادل اطلاعات در سازمان‌های دولتی و غیردولتی شتاب بیشتری به خود گرفته است. از این رو وزارت بهداشت، درمان و آموزش پزشکی به‌عنوان متولی سلامت، دارو و درمان اقدام به ایجاد شبکه‌ی نرم‌افزاری نموده و از این طریق قصد دارد سامانه‌های فعال در حوزه‌ی سلامت را با یکدیگر یکپارچه نماید. در این راستا شرکت آوای فناوری اطلاعات سلامت به عنوان اپراتور زیرساخت نرم‌افزاری یاد شده در وزارت بهداشت کار خود را در سال 1394 شروع نموده است.

با توجه به مدل توزیع شدگی، مدیریتی و نیاز به محرمیت اطلاعات تبادلی موجود در وزارت بهداشت این شرکت طرح توزیع‌شدگی را به طور کامل جهت تفکیک ترافیک‌های محلی از ملی و ایجاد فضای امن تبادل اطلاعات داخلی پشتیبانی می‌نماید.

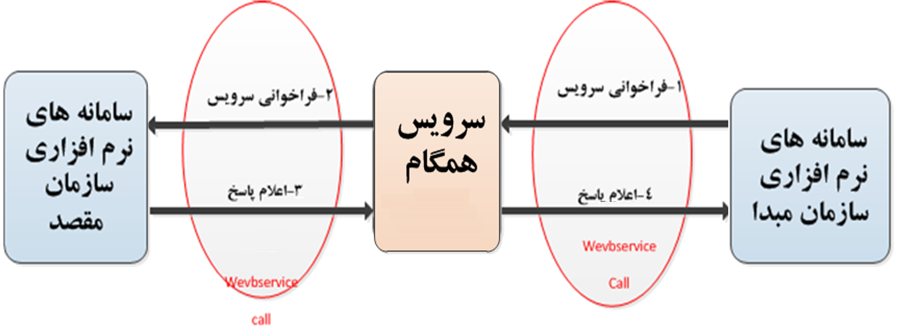
بر این اساس تبادلات اطلاعات بین سامانه‌های نرم افزاری در صورتی‌که از مرزهای مدیریتی و سازمانی فراتر رود از طریق زیر ساخت HIX انجام می‌شوند. این بدان معناست که سامانه‌های داخلی یک بیمارستان به منظور تبادل نیازی به استفاده از HIX ندارند در صورتی‌که ارتباط بین دو بیمارستان از طریق HIX صورت می‌پذیرد. بدین ترتیب با وجود زیرساخت ارتباطی و همچنین شرایط نرم‌افزاری موجود این امکان فراهم شده است تا سازمان‌های درگیر در حوزه‌ی سلامت از این زیرساخت جهت رفع نیازمندی‌های اطلاعاتی خود بهره لازم را ببرند.

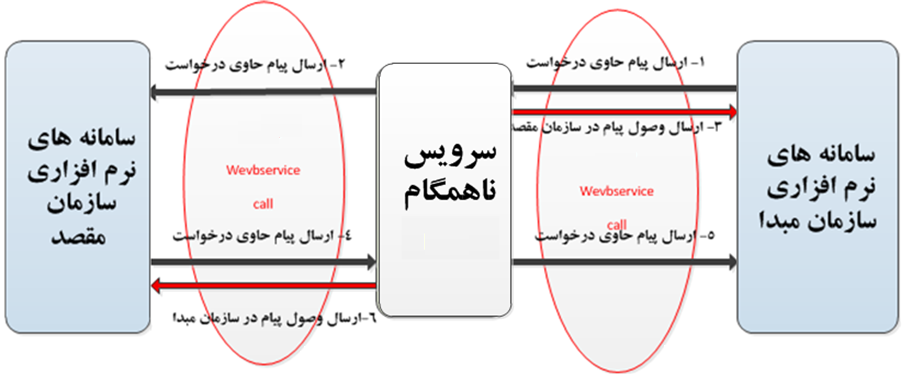
1. انتقال اطلاعات در زیرساخت HIX
   1. مدل مفهومی

زیرساخت تبادل اطلاعات سلامت (HIX) یک زیرساخت نرم‌افزاری است و به صورت یک میان‌افزا ارتباطی جهت برقراری ارتباط مکانیزه‌ی بین دو سامانه‌ی نرم‌افزاری طراحی و تولید شده است.

الگوی تبادل پیام در زیرساخت HIX می‌تواند به صورت‌های ذیل باشد:

* از دیدگاه جهتِ تبادل:
  + یک‌طرفه: به این معنا که اطلاعات از یک طرف به طرف دیگر انتقال یافته و سازمان اول منتظر پاسخ از طرف سازمان دوم نیست.
  + دوطرفه : به این معنا که اطلاعات به صورت دوطرفه انتقال یافته و طرف اول منتظر پاسخ از سمت طرف دوم است.
* از دیدگاه رعایت همزمانی تبادل:
* همگام: این ارتباط مانند استفاده از یک وب سرویس معمولی است. سامانه‌ی نرم‌افزاری مبداء در صورت فراخوانی سرویس تا زمان دریافت پاسخ مسدود می‌ماند. ارتباط همگام در اصل یک ارتباط دوطرفه است به این معنا که فراخوانی سرویس از سرویس‌دهنده نهایی حتماً با بازگشت یک مقدار همراه است. لیکن این مقدار می‌تواند به عنوان پاسخ به درخواست و یا تایید دریافت درخواست تفسیر گردد که در مورد اول تبادل دوطرفه و در مورد دوم تبادل یک طرفه فرض می‌گردد.
* ناهمگام : در این حالت سامانه‌ی ارسال کننده‌ی اطلاعات پس از ارسال درخواست به فعالیت خود ادامه داده و بلوکه نمی‌گردد. در صورتی‌که ارتباط دو طرفه باشد پاسخ به پیام درخواست به زمان دیگری موکول خواهد شد و در لحظه ارسالِ درخواست تولید و ارائه نمی‌شود.

شکل شماره 1: تبادل همگام

شکل شماره 2: تبادل دو طرفه ناهمگام

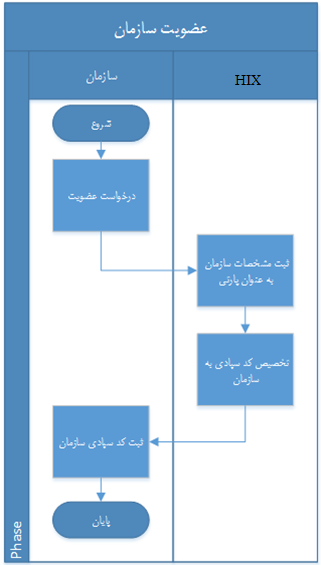
به‌عنوان مثال بیماری را درنظر بگیرید که نسخه‌ای را برای دریافت دارو نزد داروخانه می‌برد، نسخه‌ی فوق توسط داروخانه پیچیده شده و برای سازمان بیمه از طریق زیرساخت HIX ارسال می‌شود. گرفتن تأییدیه که نسخه‌ی فوق به دست سازمان بیمه رسیده است، یک ارتباط یک‌طرفه است زیرا هیچ‌گونه پردازشی بر روی نسخه هنوز صورت نگرفته است و از طرفی چنین ارتباطی همگام خوانده می‌شود چون بلافاصله پس از ارسال داروخانه منتظر دریافت تأییدیه‌ی دریافت توسط بیمه است. حال اگر سازمان بیمه تمامی نسخ موجود در یک ماه را جمع‌آوری و سپس تأییدیه‌ای برای داروخانه مبتنی بر نسخ ارسالی از سمت وی ارسال نماید، ارتباطی همچنان یک‌طرفه و ناهمگام برقرار نموده است؛ زیرا در این صورت داروخانه به ازای هر نسخه منتظر دریافت تأییدیه نیست و به فعالیت خود ادامه می‌دهد. حال شرایطی را در نظر بگیرید که برای پیچیدن نسخه، داروخانه نیازمند گرفتن تأیید اینترنتی از بیمه جهت تحویل دارو به بیمار است، در چنین شرایطی بایستی داروخانه تا زمان دریافت تأییدیه منتظر بماند و در این حالت یک ارتباط همگام (داروخانه سیستم را معطل دریافت تأییدیه‌ی اینترنتی نموده و کارها را تا زمان دریافت نتیجه‌ی تأییدیه به تعویق می‌اندازد) و همچنین ارتباطی دوطرفه خواهیم داشت زیرا از یک طرف تقاضا برای دریافت تأییدیه توسط داروخانه ارسال می‌گردد و از طرف دیگر بیمه نتیجه‌ی تأییدیه را با انجام پردازشی که بر روی داده‌های ارسالی توسط داروخانه صورت گرفته، اعلام می‌نماید. چنانچه برای تشکیل پرونده برای بیماران خاص داروخانه ملزم به انجام کار باشد، بایستی داده‌ها برای بیمه ارسال گردند و بیمه پس از بررسی‌های لازم بر روی داده‌های ارسالی نتیجه را برای داروخانه مشخص نماید (ارتباط دوطرفه) اما از آنجایی که ارائه‌ی پاسخ چند روز بعد می‌تواند اعلام شود و داروخانه در آن لحظه منتظر دریافت پاسخ نیست ارتباطی ناهمگام خواهیم داشت.

* 1. توصیف لایه انتقال

لایه‌ی انتقال در زیرساخت HIX، انتقال امن و مطمئن اطلاعات را از سامانه‌ی نرم‌افزاری مبداء به سامانه‌ی نرم‌افزاری مقصد برعهده دارد. این لایه دارای مولفه‌های ذیل است که به صورت جداگانه تشریح می‌گردند:

* + 1. متعاملین

منظور از متعامل، سازمان‌های مبداء و مقصد در تعامل هستند. این سازمان‌ها از طریق سامانه‌های نرم‌افزاری خود اقدام به برقراری ارتباط و تبادل اطلاعات می‌نمایند. تعامل بین سازمان‌ها در قالب ارائه و دریافت سرویس محقق می‌گردد.

به منظور انجام تعامل، سازمان متعامل می‌بایست به زیرساخت معرفی گردد؛ به این منظور مشخصات سازمان متعامل در دفتر ثبت زیرساخت، ثبت گردیده و یک کد کاربری به سازمان تخصیص می‌یابد. براین‌اساس سازمان می‌تواند تعاملات خود را رصد نموده و مدیریت نماید. با این کد کاربری سازمان می‌تواند سرویس‌های مورد تقاضای خود را از سازمان‌های ارائه کننده‌ی سرویس درخواست نموده و همچنین درخواست‌های تقاضای سرویس وارده به خود جهت دریافت سرویس را مدیریت نماید (تایید یا رد نموده و یا بدون پاسخ رها سازد).

سازمان می‌تواند پارامترهای کیفی ارائه‌ی سرویس را به ازای سرویس‌گیرندگان متفاوت، به‌صورت مجزا تعریف نموده و به این ترتیب مدیریت بهتری بر روی سرویس‌های قابل ارائه خود داشته باشد. به طور مثال سازمان ثبت احوال می‌تواند در زیرساخت، زمان پاسخ‌گویی به درخواست وزارت بهداشت، درمان و آموزش پزشکی را 20 ثانیه و زمان پاسخ‌گویی به درخواست سازمان غذا – دارو را 40 ثانیه تعریف نماید.

علاوه بر این سازمان می‌تواند دسترسی سازمان‌های استفاده کننده از سرویس‌های خود را به صورت موقت غیر فعال نموده و یا به‌صورت کلی ارائه‌ی سرویس را از طریق زیرساخت به شکل یک طرفه قطع نماید.

فرآیند ثبت مشخصات سازمان متعامل در زیرساخت مطابق نمودار روبرو انجام می‌پذیرد.

شکل شماره 3: فرآیند شناسایی متعاملین

* + 1. نقاط انتهایی ارتباط

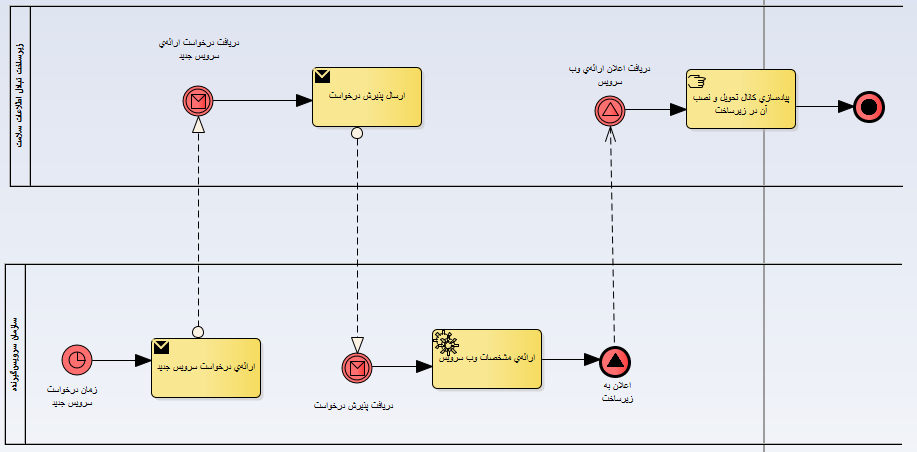
از آنجایی‌که ارتباط از طریق زیرساخت HIX به صورت ارتباط بین سامانه‌های نرم‌افزاری و به صورت مکانیزه انجام می‌شود، سامانه‌های نرم‌افزاری در سازمان‌های متعامل به عنوان نقاط انتهایی ارتباط شناخته می‌شوند.

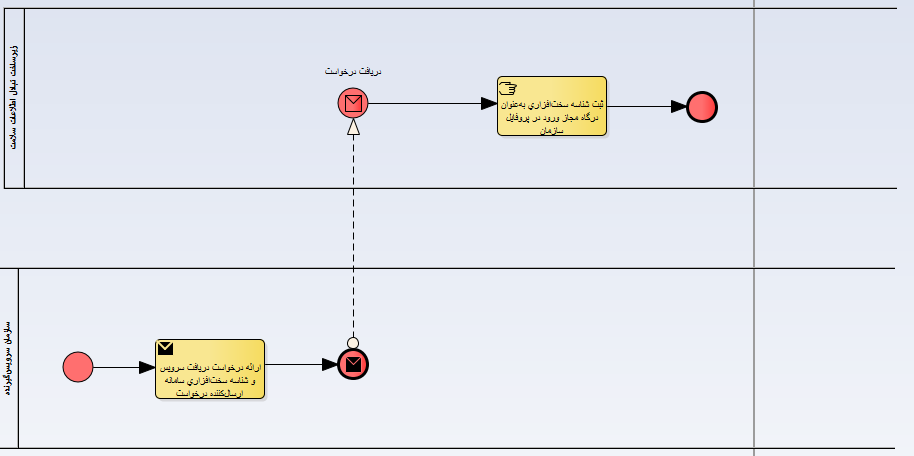
نقش سامانه‌های نرم‌افزاری براساس نقش سازمان متعامل مربوطه دارای حالت‌های ذیل است:

* سرویس‌دهنده: سامانه، سرویس‌های خود را در اختیار زیرساخت قرار داده و زیرساخت سرویس‌های مذکور را فراخوانی می‌نماید. براین اساس سامانه‌ی سرویس‌دهنده می‌تواند شرایط اتصال زیرساخت به سرویس‌های خود را به زیرساخت دیکته نماید که این شرایط می‌تواند براساس پروتکل‌های ارتباطی و سیاست‌های امنیتی متفاوت باشد. به‌طور مثال جهت اتصال به سامانه‌ی گرفتن تأییدیه اینترنتی در بیمه می‌بایست موارد ذیل رعایت گردد:
  + پیاده‌سازی ماجول سرویس‌گیرنده براساس WSDL سرویس. سرویس براساس پروتکل SOAP و به صورت RPC طراحی گردیده است (سرویس به صورت دوطرفه و همگام ارائه می‌گردد).
  + استفاده از Token سخت‌افزاری به منظور احراز هویت.
  + فراخوانی سرویس براساس URL سرویس
* سرویس‌گیرنده: سامانه، سرویس زیرساخت را فراخوانی می‌نماید. براین‌اساس زیرساخت، شرایط اتصال به سرویس‌های خود را به سرویس‌گیرنده دیکته می‌نماید. اتصال به زیرساخت تنها از طریق سرویسِ کانالِ دسترسی ممکن است. زیرساخت کلیه‌ی خدمات خود را از یک درگاه مشترک ارائه می‌نماید؛ به‌طور مثال به منظور فراخوانی هر سرویس در زیرساخت می‌بایست موارد ذیل رعایت گردد:
  + فراخوانی سرویس کانالِ دسترسی زیرساخت به صورت REST برای کلیه‌ی سرویس‌ها
  + ارائه‌ی نام سرویس و عملیات به منظور فراخوانی سرویس نهایی توسط زیرساخت
  + ارائه‌ی پارامترهای قابل ارائه به سرویس نهایی به صورت JSON
  + ارائه‌ی شناسه‌ی سخت‌افزاری مربوط به سرور سامانه‌ی نرم‌افزاری مبداء جهت احراز هویت

سازمان قبل از شروع تعامل می‌بایست سامانه(های) نرم‌افزاری خود را به زیرساخت معرفی نماید. معرفی سامانه براساس اینکه می‌خواهد نقش سرویس‌گیرنده و یا سرویس‌دهنده را ایفا نماید، متفاوت است. همچنین تعامل به صورت همگام و ناهمگام نقش تعیین‌کننده‌ای در شیوه‌ی تعریف سامانه در زیرساخت دارد:

* همگام :
  + سرویس‌دهنده: سامانه‌ی نرم‌افزاری می‌بایست سرویس خود را به صورت وب سرویس (SOAP یا REST) ارائه دهد. همچنین سیاست‌های امنیتی اتصال می‌بایست معرفی گردند. در این حالت زیرساخت با تولید یک آداپتور نرم‌افزاری به سرویس متصل می‌شود که به این آداپتور اصطلاحاً "کانال تحویل" اطلاق   
    می‌شود. (فلش با برچسب2 یا 4 از شکل شماره 1)

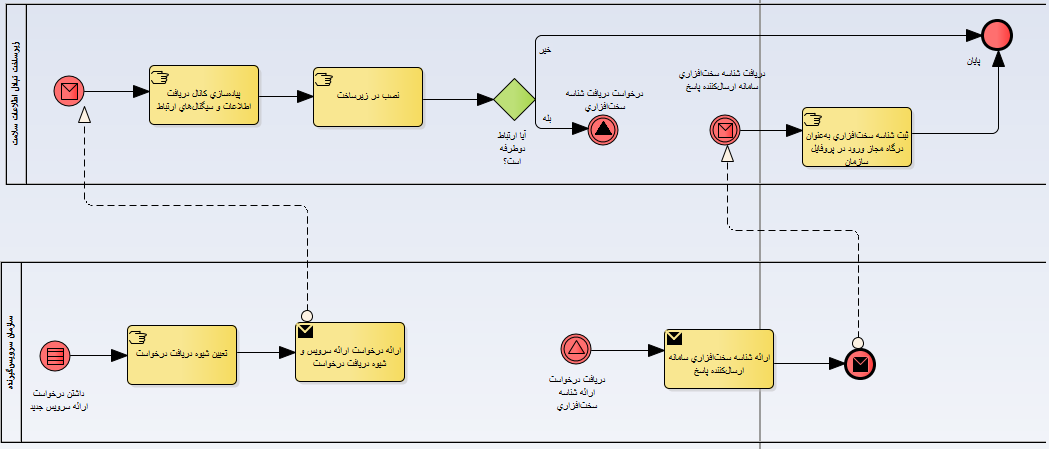
 شکل شماره 4: ارائه سرویس به‌صورت همگام

* + سرویس‌گیرنده: سامانه‌ی نرم‌افزاری می‌بایست توانایی فراخوانی کانال دسترسی زیرساخت که به صورت وب‌سرویس REST آماده‌سازی شده است را داشته باشد. به منظور رعایت سیاست‌های امنیتیِ سوییچ زیرساخت HIX، شناسه‌ی سخت‌افزاری مربوط به سروری را که سامانه‌ی نرم‌افزاری بر روی آن نصب است، می‌بایست به عنوان درگاه ورودی سازمان ارائه شده و در پروفایل مربوط به سازمان در زیرساخت ثبت گردیده باشد. (فلش با برچسب1 یا 3 از شکل شماره 1)

شکل شماره 5: دریافت سرویس به صورت همگام

* ناهمگام :
  + سرویس‌دهنده: از آنجایی‌که در سرویس ناهمگام نیازی به تولید و ارائه‌ی پاسخ در لحظه‌ی فراخوانی وجود ندارد، سامانه‌ی نرم‌افزاری سرویس‌دهنده می‌تواند درخواست را از مسیرهای مختلف دریافت نماید:
    - دریافت از طریق وب‌ سرویس: در این حالت سامانه‌ی نرم‌افزاری می‌بایست سرویس خود را به صورت وب‌سرویس (SOAP یا REST) ارائه دهد.
    - دریافت از طریق ایمیل: در این حالت سامانه‌ی نرم‌افزاری می‌بایست یک آدرس ایمیل برای دریافت درخواست‌ها به زیرساخت معرفی نماید.
    - دریافت فایل: در این حالت سرویس‌دهنده می‌بایست یک مسیر فایل را معرفی نموده تا درخواست به صورت فایل توسط زیرساخت در آن محل قرارداده شود.

در کلیه‌ی روش‌های فوق سیاست‌های امنیتی اتصال می‌بایست معرفی گردند. نهایتاً زیرساخت با تولید یک آداپتور نرم‌افزاری امکان اتصال سرویس‌دهنده به زیرساخت را فراهم می‌آورد، به این آداپتور نیز اصطلاحاً "کانال تحویل" اطلاق می‌گردد. ذکر این نکته ضروری است که شیوه‌ی دریافت درخواست از دید سامانه‌ی مبداء پنهان است (فلش با برچسب2 یا 5 از شکل شماره 2)

****شکل شماره 6: ارائه سرویس به صورت نا همگام

* + سرویس‌گیرنده: سامانه‌ی سرویس‌گیرنده در مدل ناهمگام می‌بایست دارای توانایی‌های ذیل باشد:
    - توانایی فراخوانی سرویسِ کانال دسترسی زیرساخت که به صورت وب‌سرویس REST آماده‌سازی شده است: به‌منظور رعایت سیاست‌های امنیتی سوییچ زیرساخت HIX، شناسه‌ی سخت‌افزاری مربوط به سروری که سامانه‌ی نرم‌افزاری بر روی آن نصب است می‌بایست به عنوان درگاه ورودیِ سازمان ارائه شده و در پروفایل مربوط به سازمان در زیرساخت ثبت گردیده باشد (فلش با برچسب 1 یا 4 در شکل شماره 2).
    - سرویسی ارائه نماید تا زیرساخت به منظور اعلام وصول پیام در مقصد آن را فراخوانی کند (فلش با برچسب 3 یا 6 در شکل شماره 2).

|  |
| --- |
| شکل شماره 7: دریافت سرویس به صورت ناهمگام |
| چگونگی شناسایی نقاط انتهایی ارتباط در مدل همگام در شکل 8 به تصویر کشیده شده است. همان‌طور که مشاهده می‌شود، در ابتدا سازمان درخواستی به‌منظور ارائه‌ی سرویس به زیرساخت خواهد داد. در ادامه زیرساخت مشخصات وب‌سرویسی را که توسط سازمان ارائه خواهد شد و همچنین شرایط دسترسی سازمان را دریافت خواهد نمود. سپس زیرساخت اقدام به پیاده‌سازی کانال تحویل برای سرویس وب و نصب آن در زیرساخت خواهد کرد. همچنین زیرساخت اقدام به تعریف سرویس و عملیات در زیرساخت خواهد نمود. |
| شکل شماره 8: شناسایی نقاط انتهایی ارتباط در مدل همگام  چگونگی شناسایی نقاط انتهایی ارتباط در مدل ناهمگام در شکل 9 به تصویر کشیده شده است. سازمان درخواست ارائه‌ی سرویس و مشخصات سرویس را به زیرساخت اعلام می‌دارد. با دریافت مشخصات و تأیید توسط زیرساخت، سازمان بایستی مشخصات شیوه‌ی دریافت اطلاعات را که به‌صورت وب‌سرویس، ایمیل و یا فایل خواهد بود را مشخص نماید و در اختیار زیرساخت قرار دهد. بدین ترتیب کانال تحویل اطلاعات توسط زیرساخت پیاده‌سازی و نصب گردیده؛ سپس سرویس و عملیات برای سازمان در زیرساخت تعریف و شناسه‌ی سخت‌افزاری سازمان متعامل دریافت می‌گردد. در ادامه زیرساخت شناسه‌ی فوق را به سازمان تحویل و آن را در پروفایل سازمان ثبت می‌نماید.    شکل شماره 9: شناسایی نقاط انتهایی ارتباط در مدل ناهمگام |

* + 1. زیرساخت

زیرساخت شامل کلیه‌ی مولفه‎هایی است که تبادل را بین دو سامانه ممکن می‌سازد. زیرساخت دارای دو مولفه‌ی مهم است: اولی مولفه‌ای است که سامانه‌ی نرم‌افزاری سرویس‌گیرنده به‌واسطه‌ی آن، آنرا فراخوانی می‌نماید که به این مولفه اصطلاحاً "کانال دسترسی" اطلاق می‌شود و دومی مولفه‌ای است که به‌واسطه‌ی آن سرویس سامانه‌ی نرم‌افزاری سرویس‌دهنده توسط زیرساخت فراخوان می‌شود که این مولفه اصطلاحاً "کانال تحویل" نامیده می‎شود.

* کانال دسترسی: بخشی از زیرساخت HIX است که به سامانه نرم‌افزاری مبداء این امکان را می‌دهد تا اطلاعات خود را به‌منظور انتقال به مقصد تحویل زیرساخت HIX دهد. انتقال اطلاعات به کانال دسترسی برای انتقال همگام و یا ناهمگام به یک شکل و توسط فراخوانی وب سرویس زیرساخت (REST) انجام می‌شود.
* کانال تحویل: بخشی از زیرساخت HIX است که به سامانه‌ی نرم‌افزاری مقصد این امکان را می‌دهد تا مسیر دریافت اطلاعات از زیرساخت را تعیین نماید. مشخصات این کانال براساس تبادل همگام و یا ناهمگام بین مبداء و مقصد می‌تواند متفاوت باشد:
  + ارسال همگام: درصورتی‌که سامانه‌ی مبداء، سرویس سامانه‌ی مقصد را به صورت همگام نیاز داشته باشد، سامانه‌ی مقصد می‌بایست یک سرویس‌وب (SOAP یا REST) پیاده‌سازی نموده و به زیرساخت معرفی نماید. در این صورت متناسب با سرویس ارائه شده، یک کانال دسترسی منحصر به سرویس مذکور در زیرساخت تعریف می‌گردد.
  + ارسال ناهمگام: درصورتی‌که سامانه‌ی مبداء، سرویس سامانه‌ی مقصد را به صورت ناهمگام نیاز داشته باشد، سامانه‌ی مقصد می‌تواند کانال‌های دریافت مختلفی را به زیرساخت پیشنهاد دهد. در این صورت متناسب با کانال ارائه شده یک کانال دسترسی منحصر به آن در زیرساخت تعریف می‌گردد. لازم به ذکر است که کانال تحویل در سمت سامانه‌ی مقصد از دید سامانه‌ی مبداء مخفی است و ارتباط سامانه‌ی مبداء تنها با زیرساخت و از طریق کانال دسترسی است. کانال ارائه شده می‌تواند به یکی از صورت‌های ذیل باشد:
    - Email
    - File
    - WebService (SOAP/REST)

|  |
| --- |
|  |
| شکل شماره 10: کانال‌های دسترسی و تحویل زیرساخت HIX |

1. تحویل مطمئن پیام و امنیت در زیرساخت HIX
   1. تحویل مطئمن

تحویل مطئمن پیام به این معناست که سامانه‌ی نرم‌افزاری مبداء می‌بایست این اطمینان را حاصل کند که پیام ارسالی وی به سامانه‌ی نرم‌افزاری مقصد رسیده باشد و پیام پاسخ در مدت زمان مشخصی به وی بازگردانده شود.

**ارتباط همگام**

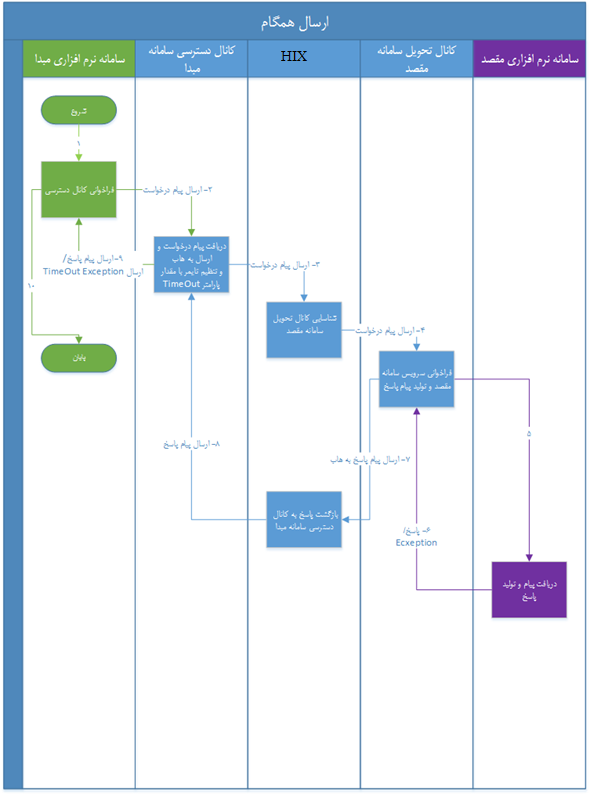
ارتباط همگام در زیرساخت به معنای فراخوانی سرویس وب همگام از سازمان سرویس‌دهنده توسط زیرساخت است. این فراخوانی باعث می‌شود تا زیرساخت تا زمان دریافت پاسخ از سرویس‌دهنده منتظر بماند. از این رو در صورتی‌که ارتباط بین دو سامانه‌ی نرم‌افزاری از نوع همگام باشد، سامانه‌ی نرم‌افزاری مبداء با فراخوانی کانال دسترسی زیرساخت باید آنقدر منتظر بماند تا پاسخ مناسب از طرف مقابل دریافت گردد. بنابراین در ارتباط همگام خاصیت تحویل مطئمن به صورت تلویحی وجود دارد.

با وجود این از آنجایی‌که زمان دریافت پاسخ در این نوع سرویس توسط سامانه‌ی نرم‌افزاری مبداء می‌تواند تعیین کننده باشد، سامانه‌ی نرم‌افزاری مبداء می‌تواند توسط پارامتر TimeOut مقدار انتظار خود را برای دریافت پاسخ به زیرساخت اعلام نماید. براین اساس هنگامی که کانال دسترسی زیرساخت توسط سامانه‌ی نرم‌افزاری مبداء فراخوانی شد، زیرساخت با انتقال پیام از طریق هاب داخلی به کانال تحویل مربوط به سامانه‌ی نرم‌افزاری مقصد، سرویس سامانه‌ی نرم‌افزاری مقصد را فراخوانی می‌کند. همزمان با این عمل، زیرساخت یک تایمر با مقدار ارائه شده توسط سامانه‌ی نرم‌افزاری مبداء (تعیین شده به صورت پارامتر در فراخوانی کانال دسترسی) مقداردهی می‌نماید. هنگامی‌که دریافت پاسخ از سامانه‌ی نرم‌افزاری مقصد بیش از زمان انتظار تعیین شده طول بکشد، زیرساخت، فراخوانی را ملغی نموده و با ایجاد یک خطای زمان (TimeOut Exception) و ارائه‌ی آن به سامانه‌ی نرم‌افزاری مبداء، این سامانه را از حالت انتظار خارج می‌نماید. شکل شماره 9 فرایند کلی ارتباط همگام از طریق زیرساخت را نمایش می‌دهد.

براساس آنچه که در شکل توصیف شده، تنها کانال‌های دسترسی و تحویلِ زیرساخت هستند که در فراخوانی همگام بلوکه   
می‌شوند و ارتباط بین کانال‌های فوق و هاب زیرساخت مبتنی بر پیام‌دهی ناهمگام بوده و مسیر رفت و برگشت پیام و پاسخ یکی نیست.

در اصل کانال دسترسی با منتظر نگاه‌داشتن فراخوانی‌کننده‌ی سرویس، ارتباط همگام را شبیه‌سازی نموده و ارتباط مستقیم و همگامی بین کانال دسترسی و کانال تحویل در زیرساخت وجود ندارد.

براین اساس کانال دسترسی با تنظیم یک تایمر می‌تواند قبل از بازگشت پاسخ پیام از کانال تحویل فراخوانی‌کننده، کانال دسترسی را رها سازد. در این حالت یک پیام مناسب از نوع Exception به فراخوانی کننده‌ی کانال دسترسی بازگشت داده می‌شود.



شکل 11 : ارتباط همگام از طریق زیرساخت

**ارتباط ناهمگام**

در صورتی‌که ارتباط بین دو سامانه‌ی نرم‌افزاری از نوع ناهمگام باشد سامانه‌ی نرم‌افزاری مبداء با فراخوانی کانال دسترسی زیرساخت، یک شماره پیام (به منظور رهگیری‌های آتی) دریافت نموده و از انتظار خارج می‌گردد. لیکن آگاهی از رسید و یا عدم رسید پیام به سامانه‌ی مقصد برای وی مهم است. همچنین حداکثر زمان انتظار برای رسیدنِ پیام ارسالی به سامانه‌ی مقصد و نیز زمان انتظار سامانه‌ی مبداء برای دریافت پاسخ از سامانه‌ی مقصد می‌تواند دو پارامتر تعیین‌کننده باشد.

به منظور پاسخ‌دهی به نیازهای مطرح شده، زیرساخت دارای مفاهیم ذیل می‌باشد:

* پیام تأییدیه رسید (Acknowledge Receipt): این پیام توسط زیرساخت تولید می‌شود و هنگامی که زیرساخت پیام را توسط کانال تحویل به سامانه‌ی مقصد تحویل می‌دهد یک پیام تأییدیه‌ی رسید تولید نموده و بازگشت به پیام ارسالی را به سامانه‌ی نرم‌افزاری مبداء ارسال می‌کند. (سامانه مبداء می‌بایست دارای کانال تحویل برای دریافت پیام مذکور باشد).

در صورتی‌که به هر دلیل کانال تحویل، در تحویل پیام به سامانه‌ی مقصد دچار مشکل شد یک پیام تأییدیه رسید با محتوی منفی برای سامانه‌ی نرم‌افزاری مبداء ارسال می‌گردد.

* زمان انتظار برای تأییدیه رسید (TimeToAck): پارامتری است که در تفاهم نامه‌ی دریافت سرویس توسط سازمان ارائه‌کننده‌ی سرویس تنظیم شده و نشان‌دهنده‌ی سطح خدمتی است که توسط ارائه‌دهنده‌ی سرویس قابل انجام است.

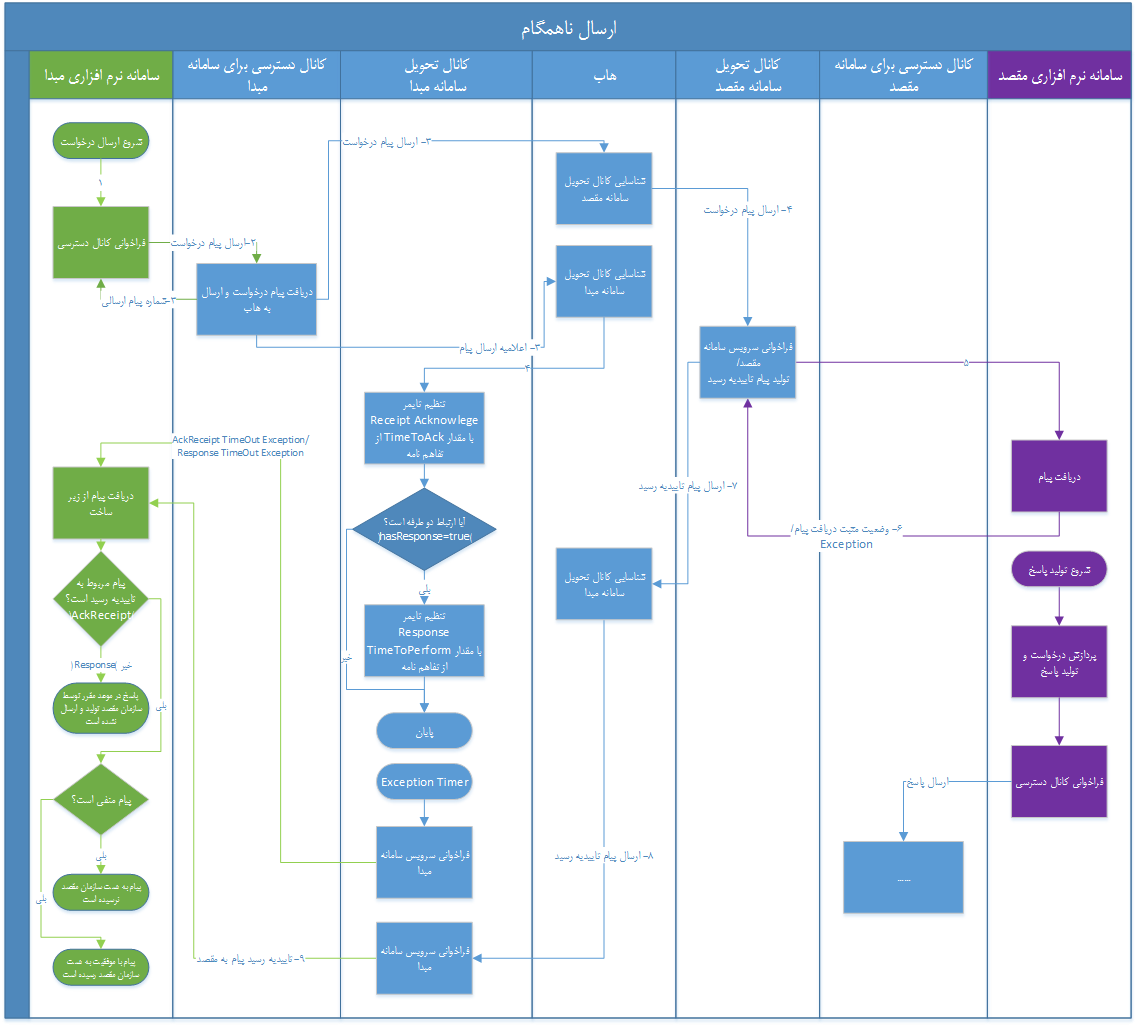
بر این اساس هنگامی‌که کانال دسترسی زیرساخت توسط سامانه‌ی نرم‌افزاری مبداء فراخوانی شد، زیرساخت با انتقال پیام از طریق هاب داخلی به کانال تحویل مربوط به سامانه‌ی نرم‌افزاری مقصد، سرویس سامانه‌ی نرم‌افزاری مقصد را فراخوانی می‌کند. همزمان با این عمل زیرساخت یک تایمر (با مقدار تعیین شده در تفاهم نامه بین دو سازمان) مقداردهی می‌نماید. هنگامی‌که تحویل پیام به سامانه‌ی نرم‌افزاری مقصد بیش از زمان انتظار تعیین شده، طول بکشد زیرساخت فراخوانی را ملغی نموده و با ایجاد یک خطای زمان (TimeOut Exception) و ارائه‌ی آن به سامانه‌ی نرم‌افزاری مبداء، سامانه‌ی نرم‌افزاری مبداء را از عدم رسید پیام به سامانه‌ی مقصد در مدت زمان تعیین شده مطلع می‌سازد.

* زمان انتظار برای بررسی درخواست و ارسال پاسخ (TimeToPerform): پارامتری است که در تفاهم نامه‌ی دریافت سرویس توسط سازمان ارائه‌کننده‌ی سرویس تنظیم شده و نشان‌دهنده‌ی سطح خدمتی است که توسط ارائه‌دهنده‌ی سرویس قابل انجام است.

براین اساس هنگامی که کانال دسترسی زیرساخت توسط سامانه‌ی نرم‌افزاری مبداء فراخوانی شد، زیرساخت پیام را طریق هاب داخلی و کانال تحویلِ مربوطه به سامانه‌ی نرم‌افزاری مقصد تحویل می‌نماید. همزمان با این عمل زیرساخت یک تایمر (با مقدار تعیین شده در تفاهم نامه‌ی بین دو سازمان) مقداردهی می‌نماید. هنگامی که دریافت پاسخ از سامانه‌ی نرم‌افزاری مقصد بیش از زمان انتظار تعیین شده طول بکشد، زیرساخت با ایجاد یک خطای زمان (TimeOut Exception) و ارائه‌ی آن به سامانه‌ی نرم‌افزاری مبداء، سامانه‌ی نرم‌افزاری مبداء را از عدم دریافت پاسخ توسط سامانه‌ی مقصد در مدت زمان تعیین شده، مطلع می‌سازد.

درصورتی‌که تعامل دوطرفه باشد، سامانه‌ی ارسال‌کننده و دریافت‌کننده‌، تعامل دو طرفه دارند؛ به این معنا که سامانه‌ی ارسال کننده می‌بایست مسیری را برای دریافت پاسخ داشته باشد. همچنین سامانه‌ی دریافت کننده‌ی پیام می‌بایست پاسخ سامانه‌ی اول را در زمان مقتضی ارسال نماید. از این رو هر یک از سازمان‌های طرف تعامل هم در نقش سرویس‌دهنده و هم در نقش سرویس‌گیرنده هستند.

در صورتی‌که تعامل یک طرفه باشد سامانه‌ی ارسال‌کننده می‌بایست مسیری را جهت دریافت وضعیت رسید پیام و یا ارسال پیامِ مربوط به خطای زمان و پاسخ ایجاد نموده تا توسط زیرساخت فراخوانی گردند.

شکل 12: ارسال درخواست به صورت ناهمگام از طریق زیرساخت

\* در شکل بالا سناریوی ارسال پاسخ همانند ارسال درخواست طی می‌گردد.

* 1. امنیت

مفهوم امنیت در زیرساخت HIX دارای لایه‌های ذیل است:

* امنیت دستیابی به کانال دسترسی برای سازمان مبداء
* کنترل دستیابی به کانال‌های تحویل مربوط به سامانه‌های سازمان مقصد
  + 1. امنیت دستیابی به کانال دسترسی

پیش‌شرط برقراری ارتباط با زیرساخت و دریافت سرویس، تعریف سازمان سرویس‌دهنده و سرویس‌گیرنده به عنوان متعامل (Party) در زیرساخت است. در هنگام تعریف سازمان، یک شناسه منحصر به فرد (PartyID) به سازمان اختصاص   
می‌یابد. سازمان سرویس‌گیرنده از این پس می‌بایست با این شناسه خود را به زیرساخت معرفی نماید(SenderID = PartyID) و همچنین می‌بایست مشخصات سازمان سرویس‌دهنده را نیز توسط شناسه‌ی آن سازمان مشخص نماید (ReciverID = PartyID).

به منظور ایجاد امنیت و همچنین کنترل ترافیک شبکه، کشور به حوزه‌های جغرافیایی و مدیریتی تقسیم‌بندی شده و به طبع آن زیرساخت نیز به‌صورت توزیع شده نصب و آماده‌ی ارائه‌ی سرویس شده است. در این توزیع‌شدگی به ازای هر حوزه، یک کانال دسترسی در زیرساخت تعریف شده است و سازمان‌ها براساس موقعیت جغرافیایی خود به یکی از این کانال‌های دسترسی متصل می‌باشند؛ این بدین معنی است که حتی در صورتی‌که یک سازمان امکان دسترسی شبکه‌ای به کانال دسترسی دیگر را داشته باشد زیرساخت از ارائه‌ی سرویس به سازمان جلوگیری می‌نماید.

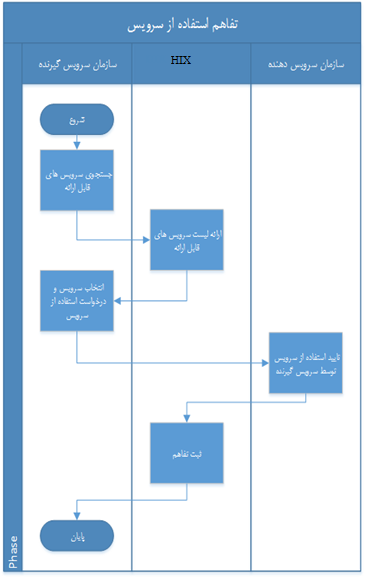
دریافت و ارائه‌ی سرویس نیز در زیرساخت تنها از طریق سامانه‌های نرم‌افزاری سازمان‌ها صورت می‌پذیرد و زیرساخت رابط گرافیکی جهت ارائه به کاربران در اختیار ندارد. دریافت سرویس از زیرساخت توسط کانال دسترسی انجام شده که به ازای کلیه‌ی سرویس‌های درخواستی سازمان یکتاست؛ این بدین معنی است که سامانه‌ی نرم‌افزاری سرویس‌گیرنده کلیه‌ی سرویس‌های خود را تنها با فراخوانی یک کانال دسترسی درخواست می‌نماید و نیازی به فراخوانی سرویس‌های مختلف ندارد.

به منظور تسهیل در امر شناسایی سامانه‌های نرم‌افزاری مبداء (فراخوانی‌کننده‌ی کانال دسترسی)، تنها شناسه‌ی سخت‌افزاری مربوط به سروری که نرم‌افزارهای سازمان مبداء بر روی آن قرار دارند، می‌بایست به عنوان نقاط انتهایی ارتباط به زیرساخت معرفی گردند.

لازم به ذکر است که زیرساخت نیازی به شناسایی سامانه‌های نرم‌افزاری سازمان مقصد به این شکل ندارد زیرا به ازای هر یک از سرویس‌های ارائه شده توسط آنها اقدام به شخصی‌سازی کانال تحویل می‌نماید. این شخصی‌سازی می‌تواند حتی در سطح پیاده‌سازی سیاست‌های امنیتی برای هر سازمان و برای هر سرویس به طور جداگانه انجام شود.

شکل 13 – رابطه بین سرویس‌های ارائه شده توسط سامانه مقصد و کانال تحویل مربوط به زیرساخت

* + 1. امنیت دستیابی به سرویس‌های مقصد

سازمان‌های سرویس‌دهنده سرویس‌های خود را بر روی زیرساخت منتشر می‌نمایند. آیا این بدان معنی است که زیرساخت توان اتصال به سرویس‌های سرویس‌دهنده را داراست و در صورتی‌که هر سازمان دیگری به زیرساخت دسترسی داشته باشد لاجرم به سرویس‌های منتشر شده بر روی زیرساخت نیز دسترسی دارد؟

جواب این سئوال خیر است؛ سازمان سرویس‌گیرنده می‌بایست برای دریافت از سازمان سرویس‌دهنده دارای تفاهم نامه‌ی ثبت‌شده در زیرساخت باشد. یکی از کاربردهای تفاهم نامه در زیرساخت، ایجاد شرایط کنترل دسترسی به سرویس‌های سازمان سرویس‌گیرنده است. از این رو سازمان سرویس‌گیرنده می‌بایست درخواست استفاده از سرویس را به سازمان سرویس‌دهنده ارسال نموده و پس از تأیید سازمان سرویس‌دهنده، تفاهم در زیرساخت ثبت می‌شود و از آن لحظه به بعد سرویس در دسترسِ سازمان سرویس‌گیرنده قرار خواهد داشت. سازمان سرویس‌دهنده همچنین می‌تواند دسترسی سازمان سرویس‌گیرنده را به صورت موقت قطع نماید.

از این به بعد دو سازمان سرویس‌دهنده و سرویس‌گیرنده می‌توانند لاگ های مربوط به تبادل اطلاعاتشان را به ازای هر تفاهم نامه در زیرساخت مشاهده نمایند.

شکل شماره 14 : فرآیند ثبت تفاهم‌نامه‌ی استفاده از سرویس

1. فراخوانی کانال دسترسی

سرویس‌های همگام و ناهمگام در سمت زیرساخت HIX توسط یک وب سرویس عمومی پیاده‌سازی شده‌اند. پیاده‌سازی این وب سرویس بصورت REST می‌باشد. وب سرویس مذکور در آدرسی با ساختار زیر قرار خواهد گرفت که در آن SAPAAD -SERVER-URL مربوط به محل نصب سرویس REST زیرساخت می‌باشد:

http://<SAPAAD-SERVER\_URL>:8888/dispatch

توجه :

1. فراخوانی وب سرویس مذکور بصورت POST (http POST) می‌باشد.
2. پارامتر‌های json بایستی در request body اضافه گردند.
3. پاسخ آماده شده برای سرویس پست در قالب فرمت UTF-8 آماده می‌گردد.
4. فقط بخش‌های مربوط به Payload در ورودی متدها بایستی تکمیل گردد . اقلام دیگر به‌عنوان متغیرهای ثابت کنترلی می‌باشند.
   1. فراخوانی کانال دسترسی به‌صورت همگام

کانال دسترسی به‌صورت یک وب سرویس REST در اختیار استفاده‌کنندگان قرار دارد. براین اساس فراخوانی کانال دسترسی به صورت ارسال پیام به وب سرویس و دریافت پاسخ از آن است. همان‌گونه که ذکر شد، در مدل همگام به محض فراخوانی کانال دسترسی، پیام به کانال تحویل ارسال شده و سرویس سازمان مقصد فراخوانی می‌گردد. بر این اساس فراخوانی سرویس دسترسی یا همراه با خطا خواهد بود و یا مقدار بازگشتی از سرویس مقصد در لحظه‌ی فراخوانی کانال دسترسی بازگشت داده می‌شود. فرمت اطلاعات ارسالی و دریافتی به/از وب سرویس کانال دستری در ادامه آمده است:

* + 1. ارسال پیام به وب سرویس کانال دسترسی

به منظور فراخوانی کانال دسترسی به صورت همگام می‌بایست پارامترهای ذیل به وب سرویس کانال دسترسی ارسال گردند:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ردیف | نام پارامتر | نوع | اجباری؟ | توضیح |
| 1 | senderId | String | اجباری | کد سازمان مبداء. شناسه‌ای است که به عنوان PartyID در زمان ثبت مشخصات متعامل به آن تخصیص داده می‌شود. |
| 2 | receiverId | String | اجباری | کد سازمان مقصد. شناسه‌ای است که به عنوان PartyID در زمان ثبت مشخصات متعامل به آن تخصیص داده می‌شود. |
| 3 | service | String | اجباری | نام سرویس در سازمان مقصد |
| 4 | action | String | اجباری | نام عملیات مربوط به سرویس در سازمان مقصد |
| 5 | version | String | اجباری | شماره نسخه مربوط به سرویس |
| 6 | replyToConversationId | String | اختیاری | شماره پیامی است که این پیام به‌صورت پاسخ به آن ارسال می‌شود |
| 7 | timeout | String | اختیاری | مقدار زمان تاخیر انتظار برای دریافت سرویس را به ثانیه تعیین می‌نماید. این مقدار می‌بایست برابر و یا بزرگتر از مقدار تعیین شده در تفاهم‌نامه دریافت سرویس بین سازمان مبداء (سرویس‌گیرنده) و سازمان مقصد (سرویس‌دهنده) باشد. در صورت عدم ارائه‌ی این پارامتر مقدار تعیین شده در تفاهم، توسط زیرساخت در نظر گرفته می‌شود. |
| 8 | payload | String | اختیاری | حاوی مقادیری است که به صورت لیستی از پارامترها و یا سند به سرویس سازمان مقصد ارسال می‌گردد. این پارامتر بسته به عملیات سرویس متفاوت مقداردهی شده و شرح جزئیات تکمیل آن در اسناد مربوط به هر سرویس آورده می‌شود. |
| 9 | payloadClass | String | اجباری | نشان‌دهنده‌ی ساختاری است که محتویات payload براساس آن مقداردهی شده و اعتبار سنجی می‌گردد.  این پارامتر بسته به عملیات سرویس متفاوت مقداردهی شده و شرح جزئیات تکمیل آن در اسناد مربوط به هر سرویس آورده می‌شود.  در صورتی‌که سرویس دارای مقدار ورودی نیست این پارامتر می‌بایست با مقدار ذیل تکمیل گردد:  ix.dispatch.client.EmptyRequest |
| 10 | plainId | String | اجباری | شناسه آشکار |
| 11 | encryptedId | String | اجباری | شناسه امن |

تکمیل پارامترها به منظور فراخوانی سرویس به‌صورت همگام در قالب json به‌صورت زیر است:

##request:

{

"payload" : {

// مقادیر پارامترهای فراخوانی سرویس مقصد

},

"payloadClass" : "نام کلاس مربوط به مورد کاربردی",

"senderId" : "کد سازمان مبداء",

"receiverId" : "کد سازمان مقصد",

"service" : "نام سرویس در سازمان مقصد",

"action" : "نام عملیات در سرویس سازمان مقصد",

"version" : "نسخه مربوط به سرویس",

"replyToConversationId" : "شماره پیام مبداء",

"timeout" : "مقدار زمان انتظار برای دریافت پاسخ",

"plainId" :" شناسه آشکار ",

"encryptedId" :"شناسه امن"

}

\* اطلاعات شناسه‌های آشکار و امن در بخش امنیت همین سند آمده است.

* + 1. بازگشت پاسخ از وب سرویس کانال دسترسی

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ردیف | نام پارامتر | نوع | اجباری؟ | توضیح |
| 1 | type | String | اجباری | نوع پاسخ را تعیین می‌کند که شامل مقادیر ذیل است:   * ORDINARY : در صورتی که فراخوانی سرویس به صورت موفق انجام شده باشد. در این صورت payload حاوی مقادیر بازگشتی است. * EXCEPTION : در صورتیکه فراخوانی سرویس به صورت ناموفق انجام شده باشد. در این صورت payload حاوی محتویات خطای رخ داده است. |
| 2 | conversationId | String | اجباری | شماره منحصر به فردی است که زیرساخت به هر فراخوانی نسبت می‌دهد. (شماره پیام) |
| 3 | payload | String | اختیاری | محتویات این پارامتر بر اساس پارامتر type تعیین می‌شود. بر این اساس این پارامتر حاوی مقادیر بازگشتی از فراخوانی سرویس و یا خطای رخ داده است. |
| 4 | payloadClass | String | اجباری | نشان‌دهنده‌ی ساختاری است که محتویات payload براساس آن مقداردهی شده است:   * درصورتی‌که فراخوانی سرویس با موفقیت انجام شود این پارامتر دارای نام کلاسی خواهد بود که مرتبط با عملیات فراخوانی شده است. نام کلاس مرتبط در مستند مربوط به هر سرویس به صورت جداگانه ارائه می‌گردد. * در صورتی‌که فراخوانی سرویس همراه با خطا باشد این پارامتر دارای مقدار ذیل است:   ix.dispatcher.commons.ExceptionPayload |

در صورت فراخوانی موفق، مقادیر بازگشتی از وب سرویس کانال دسترسی در قالب json به‌صورت زیر است:

##response:

{

"payload" : {

//مقادیر بازگشتی

}

},

"payloadClass" : "نام کلاس مربوط به مورد کاربری",

"type" : "ORDINARY",

"conversationId" : "شماره پیام"

}

* 1. فراخوانی کانال دسترسی بصورت نا همگام

در مدل ناهمگام سامانه‌ی نرم‌افزاری سازمان مبداء در انتظار فراخوانی سرویس سازمان مقصد توسط زیرساخت قرار نمی گیرد. براین اساس پس از فراخوانی موفق زیرساخت یک شماره پیام به سامانه نرم‌افزاری سازمان مبداء بازگشت داده و فراخوانی کانال دسترسی تکمیل می‌گردد. در ادامه‌ی فرآیند، زیرساخت نتیجه‌ی فراخوانی سرویس سازمان مقصد را از طریق کانال تحویل به سامانه‌ی نرم‌افزاری سازمان مبداء اطلاع می‌دهد.

علاوه بر این در مدل ناهمگام این امکان وجود دارد تا پیام به صورت همزمان برای سازمان‌های مختلف ارسال گردد؛ یعنی سازمان مبداء از زیرساخت درخواست می‌نماید تا سازمان های مقصد بیش از یک عدد باشند، در این صورت زیرساخت مجاب می‌گردد تا سرویس مقصد در سازمان‌های مختلفی را فراخوانی نموده و پیام دریافتی را به آنها تحویل نماید.

در ادامه فراخوانی کانال دسترسی به صورت ناهمگام در دو حالت ذیل توصیف می‌گردد:

* ارسال پیام به یک مقصد
* ارسال پیام به چند مقصد
  + 1. ارسال پیام به یک مقصد

در این مدل، گیرنده‌ی پیام تنها یک سازمان مقصد است.

* + 1. ارسال پیام به وب سرویس کانال دسترسی

| ردیف | نام پارامتر | نوع | اجباری؟ | توضیح |
| --- | --- | --- | --- | --- |
| 1 | senderId | String | اجباری | کد سازمان مبداء |
| 2 | senderService | String | اجباری | مشخصات سرویسی است که سامانه‌ی نرم‌افزاری جهت دریافت نتیجه ارسال به زیرساخت معرفی می‌نماید |
| 3 | senderVersion | String | اجباری |
| 4 | receiverId | String | اجباری | کد سازمان مقصد |
| 5 | service | String | اجباری | نام سرویس در سازمان مقصد |
| 6 | action | String | اجباری | نام عملیات مربوط به سرویس در سازمان مقصد |
| 7 | version | String | اجباری | شماره نسخه مربوط به سرویس |
| 8 | replyToConversationId | String | اختیاری | شماره پیامی است که این پیام به‌صورت پاسخ به آن ارسال می‌شود |
| 9 | hasResponse | Boolan | اجباری | تعامل یک طرفه و یا دو طرفه را به زیرساخت اعلام می‌نماید. |
| 10 | timeToAck | Number | اختیاری | مقدار زمان تاخیر انتظار برای آگاهی از وضعیت ارسال پیام به سازمان مقصد را به ثانیه تعیین می‌نماید.  این مقدار می‌بایست برابر و یا بزرگتر از مقدار تعیین شده در تفاهم دریافت سرویس بین سازمان مبداء (سرویس گیرنده) و سازمان مقصد (سرویس‌دهنده) باشد.  در صورت عدم ارائه این پارامتر، مقدار تعیین شده در تفاهم نامه توسط زیرساخت برای فراخوانی در نظر گرفته می‌شود. |
| 11 | timeToPerform | Number | اختیاری | در تعاملات دو طرفه مقدار زمان تاخیر انتظار برای دریافت پیام از سازمان مقصد در پاسخ به پیام ارسالی را به ثانیه تعیین می‌نماید.  این مقدار می‌بایست برابر و یا بزرگتر از مقدار تعیین شده در تفاهم‌نامه دریافت سرویس بین سازمان مبداء (سرویس گیرنده) و سازمان مقصد (سرویس‌دهنده) باشد.  در صورت عدم ارائه‌ی این پارامتر مقدار تعیین شده در تفاهم نامه توسط زیرساخت برای فراخوانی در نظر گرفته می‌شود. |
| 12 | payload | String | اختیاری | حاوی مقادیری است که به صورت لیستی از پارامترها و یا سند به سرویس سازمان مقصد ارسال می‌گردد. این پارامتر بسته به عملیات سرویس متفاوت مقداردهی شده و شرح جزئیات تکمیل آن در اسناد مربوط به هر سرویس آورده می‌شود. |
| 13 | payloadClass | String | اجباری | نشان‌دهنده‌ی ساختاری است که محتویات payload براساس آن مقداردهی شده و اعتبارسنجی می‌گردد.  این پارامتر بسته به عملیات سرویس متفاوت مقداردهی شده و شرح جزئیات تکمیل آن در اسناد مربوط به هر سرویس آورده می‌شود.  در صورتی‌که سرویس دارای مقدار ورودی نیست این پارامتر می‌بایست با مقدار ذیل تکمیل گردد:  ix.dispatch.client.EmptyRequest |
| 14 | plainId | String | اجباری | شناسه آشکار |
| 15 | encryptedId | String | اجباری | شناسه امن |

فراخوانی سرویس کانال دسترسی به صورت ناهمگام در قالب json به‌صورت زیر است:

##request:

{

"payload" : {

// مقادیر پارامترهای فراخوانی سرویس مقصد

},

"payloadClass" : "نام کلاس مربوط به مورد کاربردی",

"senderId" : "کد سازمان مبداء",

"senderService" :"نام سرویس سازمان مبداء ",

"senderVersion" :" نسخه مربوط به سرویس سازمان مبداء ",

"receiverId" : "کد سازمان مقصد",

"service" : "نام سرویس در سازمان مقصد",

"action" : "نام عملیات در سرویس سازمان مقصد",

"version" : "نسخه مربوط به سرویس سازمان مقصد",

"replyToConversationId" : "شماره پیام مبداء",

"hasResponse" : true/false,

"timeToAck" :"مدت زمان انتظار برای اعلام وضعیت پیام ارسال شده",

"timeToPerform" :"مدت زمان انتظار برای دریافت پیام پاسخ در تعاملات دو طرفه",

"plainId" :" شناسه آشکار ",

"encryptedId" :"شناسه امن",

}

* + - 1. بازگشت پاسخ از وب سرویس کانال دسترسی

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ردیف | نام پارامتر | نوع | اجباری؟ | توضیح |
| 1 | type | String | اجباری | نوع پاسخ را تعیین می‌کند که شامل مقادیر ذیل است:   * ORDINARY : در صورتی که فراخوانی سرویس به صورت موفق انجام شده باشد. در این صورت payload حاوی مقادیر بازگشتی است. * EXCEPTION : در صورتیکه فراخوانی سرویس به صورت ناموفق انجام شده باشد. در این صورت payload حاوی محتویات خطای رخ داده است. |
| 2 | conversationId | String | اجباری | شماره‌ی منحصر به فردی است که زیرساخت به هر فراخوانی نسبت می‌دهد. |
| 3 | payload | String | اجباری | محتوی مقدار ثابت received |
| 4 | payloadClass | String | اجباری | محتوی مقدار ثابت java.lang.String |

سند پاسخ آدرس پست در قالب json به‌صورت زیر است:

##response:

{

"payload" : "received",

"payloadClass" : "java.lang.String",

"type" : "ORDINARY",

"conversationId" : "شماره پیام ارسالی"

}

* + 1. تاییدیه رسید پیام به مقصد در ارسال ناهمگام

زمانی‌که پیام توسط زیرساخت به مقصد ارسال می‌شود سرویس مقصد فراخوانی شده و سعی در تحویل پیام به سازمان مقصد   
می‌کند. در این فرآیند حالت‌های ذیل اتفاق می‌افتد:

* پیام به دست سرویس سازمان مقصد می‌رسد: در این صورت زیرساخت با ارسال یک پیام، به ارسال کننده‌ی پیام اطلاع می‌دهد که پیام شما به مقصد تحویل شد. فرمت پیام مذکور که به آن Positive Acknowledge Receipt اطلاق می‌شود مانند ذیل است:

## response

{

"reason" : null,

"status" : true,

"type" : "RECEIPT",

"ackToConversationId" : " شماره پیام ارسال شده"

}

* پیام به دست مخاطب نمی‌رسد: در این صورت زیرساخت با ارسال یک پیام، به ارسال‌کننده‌ی پیام اطلاع می‌دهد که پیام شما به مقصد تحویل نشد. فرمت پیام مذکور که به آن Negative Acknowledge Receipt اطلاق می‌شود، به‌شکل کلی زیر است:

## response

{

" reason" : {

"exceptionCode" : "exceptionCode",

"exceptionMessage" : "exceptionMessage",

"exceptionStackTrace" : " exceptionStackTrace ",

},

"status" : false,

"type" : "RECEIPT",

"ackToConversationId" : " شماره پیام ارسال شده"

}

\* نکته : فرآیند اعلام Positive Acknowledge و Negative Acknowledge در ارسال پیام به یک سازمان و یا چند سازمان به دلیل استقلال پیام ها از یکدیگر از دیدگاه زیرساخت و یا سامانه‌ی نرم‌افزاری مبداء هیچ تفاوتی ندارد.

* + 1. خطاهای زمانی در ارسال ناهمگام

همان‌طور که پیشتر گفته شد در ارسال ناهمگام دو زمانبندی وجود دارد:

* اولی مربوط به حداکثر انتظار فرستنده پیام برای تحویل پیام به مقصد می‌باشد. این پارامتر که در تفاهم‌نامه‌ی بین سازمان سرویس‌گیرنده و سازمان سرویس‌دهنده تعیین می‌شود زیرساخت را متعهد   
  می‌کند تا زمان‌بندی ارسال را تعیین کرده و پس از زمان سپری شده، فرستنده‌ی پیام را از وقوع این رخداد مطلع نماید.

## response

{

" reason" : {

"exceptionCode" : "timeout.occurred",

"exceptionMessage" : "the timeout has occurred",

"exceptionStackTrace" : null,

},

"status" : false,

"type" : "RECEIPT",

"ackToConversationId" : " شماره پیام ارسال شده"

}

\* در صورتی‌که زیرساخت قبل از سررسیدن زمان تایمر، پیام Ack (مثبت/منفی) صادر نماید تایمر غیر فعال می‌شود.

\* در صورتی‌که این تایمر قبل از ارسال تأییدیه‌های رسید (مثبت یا منفی) فعال گردد، پیام های مذکور Mask شده و به اطلاع فرستنده‌ی پیام نمی‌رسند.

* دومی مربوط به حداکثر زمان انتظار فرستنده‌ی پیام برای دریافت پاسخ از مقصد است. این پارامتر در تفاهم‌نامه‌ی بین سازمان سرویس‌گیرنده و سازمان سرویس‌دهنده تعیین می‌شود. در صورتی‌که فرستنده، پارامتر hasResponse = true را در هنگام ارسال پیامِ به کانال دسترسی تنظیم نماید، زیرساخت را متعهد می‌کند تا زمان‌بندی دریافت پاسخ را تعیین کرده و پس از زمان سپری شده فرستنده‌ی پیام را از وقوع این رخداد مطلع نماید.

## response

{

" reason" : {

"exceptionCode" : "timeout.occurred",

"exceptionMessage" : "the timeout has occurred",

"exceptionStackTrace" : null,

},

"status" : false,

"type" : "PERFORMED",

"ackToConversationId" : " شماره پیام ارسال شده"

}

\* از دید زیرساخت هر پیامی که توسط گیرنده‌ی پیام به سمت فرستنده ارسال می‌شود و فیلد replyToConversationId آن به شماره پیامی اشاره کند که تایمر پاسخ برای آن فعال شده است، در صورتی‌که پیام قبل از سررسیدن زمان تایمر دریافت شود، زیرساخت با فرض اینکه پاسخ پیام ارسالی دریافت شده، تایمر را غیرفعال می‌کند.

\* در صورتی‌که این تایمر فعال گردد از ارسال پاسخ توسط گیرنده به فرستنده جلوگیری نمی‌شود.

* 1. مدیریت خطا به صورت عمومی

از آنجایی‌که زیرساخت HIX رابط بین سرویس‌گیرنده و سرویس‌دهنده است دو دسته خطا ممکن است در ارتباط رخ دهد:

* خطاهای ایجاد شده توسط کانال دسترسی : این خطاها مربوط به زیرساخت بوده و به سرویس‌دهنده‌ی نهایی مربوط   
  نمی‌گردند. خطاهایی که توسط کانال دسترسی تولید می‌شوند و عموماً دارای طبقه بندی ذیل هستند:
  + خطای نقص در پارامترهای فراخوانی سرویس کانال دسترسی
  + خطای امنیتی دسترسی به زیرساخت و یا سرویس نهایی
  + خطای عدم برپایی سرویس‌دهنده نهایی
  + خطای زمانی
  + خطای داخلی
* خطاهای ایجاد شده توسط سرویس‌دهنده‌ی نهایی : این خطاها مربوط به زیرساخت نبوده و به سرویس‌دهنده‌ی نهایی (سرویس سازمان مقصد) مربوط می‌گردند. در این حالت زیرساخت در فراخوانی سرویس مقصد دچار خطا می‌شود.

در این سند خطاهای زیرساخت تشریح شده و به منظور آگاهی از خطاهای تولید شده توسط سرویس‌دهنده‌ی نهایی می‌بایست به اسناد مربوط به سرویس مذکور مراجعه گردد.

* + 1. بروز خطا در فراخوانی کانال دسترسی

در صورت بروز خطا در فراخوانی سرویسِ کانال دسترسی، پیام زیر تولید شده و بازگشت داده می‌شود:

## response

{

"payload" : {

"exceptionCode" : "exceptionCode",

"exceptionMessage" : "exceptionMessage",

"exceptionStackTrace" : " exceptionStackTrace ",

},

"payloadClass" : "ix.dispatcher.commons.ExceptionPayload",

"type" : "EXCEPTION",

"conversationId" : "شماره پیام"

}

در صورتی‌که زیرساخت به صورت ارسال پیام به چند سازمان فراخوانی گردد، ممکن است برخی از پیام ها دچار خطا بوده و مابقی بدون اشکال باشند؛ از این رو خطا داخل payload مربوط به هر سازمان مقصد اعلام می‌گردد. به طور مثال پاسخ زیر در فراخوانی دو مقصد ارائه گردیده است. ارسال پیام به سازمان 1 دچار خطا شده و ارسال پیام به سازمان 2 بدون مشکل انجام شده است:

##response:

{

"payload" : {

"conversationsDetails" : [ {

"receiverId" : "کد سازمان مقصد1",

"response" : {

"payload" : {

"exceptionCode" : "exceptionCode",

"exceptionMessage" : "exceptionMessage",

"exceptionStackTrace" : " exceptionStackTrace ",

},

"payloadClass" : "ix.dispatcher.commons.ExceptionPayload",

"type" : "EXCEPTION",

"conversationId" : "شماره پیام ارسالی به سازمان 1 "

}

},

{

"receiverId" : "کد سازمان مقصد2",

"response" : {

"payload" : "received",

"payloadClass" : "java.lang.String",

"type" : "ORDINARY",

"conversationId" : "شماره پیام ارسالی به سازمان 2 "

}

} ]

},

"payloadClass" : "ix.dispatch.client.multirecipient.MultiRecipientResponse",

"type" : "MULTI\_RECIPIENT",

"conversationId" : null

}

همانگونه که مشخص شد، در صورت بروز خطا، Payload با یک Object از نوع ExceptionPayload تکمیل شده و ارائه می‌گردد. exceptionCode یا کد خطا یک رشته خواهد بود و براساس جدول ذیل تفسیر می‌گردد.

| **ردیف** | **کد خطا (exceptionCode)** | **تفسیر خطا (تفسیر متن exceptionMessage)** |
| --- | --- | --- |
| 1 | required.fields.not.found | یکی از فیلد هایِ اجباری پیامِ درخواست ارسال نشده است. |
| 2 | system.error | ایراد سیستمی |
| 3 | agreement.not.found | سازمان سرویس‌گیرنده تفاهم‌نامه‌ی فعالی با سازمان سرویس‌دهنده بر روی سرویس و عملیات درخواستی ندارد. |
| 4 | sender.not.corresponding.dispatcher | سازمان سرویس‌گیرنده از کانال دسترسی مناسب متصل نشده است |
| 5 | micro.server.unreachable | کانال تحویل سازمان سرویس‌دهنده موقتاً در دسترس نیست. بعدا تلاش نمایید |
| 6 | security.pass.gen.invalid | ترکیب فیلدهای 'encryptedId, plainId, senderId' نامعتبر است |
| 7 | agreement.time.violated | از زمان تفاهم‌نامه گذشته است |
| 8 | agreement.max.request.violated | تعداد درخواست‌ها از تعداد مشخص شده در تفاهم‌نامه بیشتر است |
| 9 | agreement.temporary.deactivate | تفاهم‌نامه موقتاً توسط سازمان سرویس‌دهنده غیرفعال شده است |
| 10 | timeout.occurred | خطای زمانی رخ داده است. |
| 11 | receiverId.xor.receiverIds | شما می‌توانید به صورت همزمان تنها ارسال به یک مقصد یا ارسال به چند مقصد را انجام دهید. |
| 12 | this.action.only.one.receiverId | این عملیات تنها می‌تواند در حالت ارسال پیام به یک مقصد استفاده شود. |

لازم به ذکر است که جدول فوق مربوط به ایراداتی است که بعد از فراخوانی موفقِ کانال دسترسی است و توسط کانال دسترسی صادر می‌گردد. ایرادات مربوط به فراخوانی که مربوط به کدهای خطای پروتکل HTTP می‌شوند، در این جدول نیامده است و فرض می‌گردد که کدهای مذکور واضح و قابل‌درک هستند.

* 1. امنیت در فراخوانی سرویس

همانگونه که در بخش‌های قبل ذکر شد سازمان‌ها می‌بایست به عنوان متعامل در زیرساخت معرفی گردیده و نقاط انتهایی ارتباط نیز به صورت یک شناسه‌ی سخت‌افزاری برای سامانه‌های نرم‌افزاری مبداء، به زیرساخت ارائه گردد. از این رو می‌بایست در فرآیند ثبت اطلاعات متعاملین، شناسه‌ی سخت‌افزاری ذکر شده توسط سازمان به زیرساخت HIX ارائه شده تا در زیرساخت ثبت و تنظیم گردد. از این رو مراحل طی‌شونده برای فراخوانی سرویس از زیرساخت HIX دارای مراحل ذیل است:

* ثبت مشخصات سازمان به عنوان Party در زیرساخت
* صدور شناسه‌ی سخت‌افزاری مربوط به سخت‌افزاری که سامانه(های) نرم‌افزاری مبداء بر روی آن نصب شده است.
* پیاده‌سازی کتابخانه (PS API) Partner Security API
* یکپارچه‌سازی PS API توسط سازمان
* نوع پروتکل‌ها

كتابخانه PS API شامل فايل‌هاي زير مي‌باشد كه در اختيار تأمين‌كننده قرار مي‌گيرد.

|  |  |  |
| --- | --- | --- |
| رديف | شرح فايل | نام فايل |
| 1 | فايل مربوط به دستگاه هاي 32 بيتي براي استفاده در Windows و برنامه‌هاي غير از Java مانند Delphi ، C و ... | BPJWSSec32.dll |
| 2 | فايل مربوط به دستگاه هاي 64 بيتي براي استفاده در Windows و برنامه‌هاي غير از Java مانند Delphi ، C و ... | BPJWSSec64.dll |
| 3 | فايل مربوط به دستگاه هاي 32 بيتي براي استفاده در Windows و برنامه‌هاي Java | JniBPJWSSec32.dll |
| 4 | فايل مربوط به دستگاه هاي 64 بيتي براي استفاده در Windows و برنامه‌هاي Java | JniBPJWSSec64.dll |
| 5 | فايل مربوط به دستگاه هاي 64 بيتي براي استفاده در Linux و برنامه‌هاي غير از Java | libBPJWSSec64.so |
| 6 | فايل مربوط به دستگاه هاي 64 بيتي براي استفاده در Linux و برنامه‌هاي Java | libJniBPJWSSec64.so |
| 7 | فايل جاوا براي كار با فايل هاي رديف 3 ، 4 و 6  اين فايل بايستي در برنامه كلاينت و در package زير قرار گيرد  bpj.secdep.secutil  فايل هاي dll نيز بايستي در مسير پروژه و يا كنار jar فايل اجرايي برنامه قرار گيرند  \*\* اجرای برنامه‌هاي جاوا و به‌کارگیری این کتابخانه در نسخه جاری برای JDK 1.7+ امکان پذیر می‌باشد | JniBPJWSSec.java |
| 8 | فايل مربوط به برنامه‌هايی که بر روی IIS سرور استقرار می‌یابند. به همراه این کتابخانه یک پوشه حاوی 4 dll دیگر قرار دارد که این پوشه بایستی در کنار این کتابخانه قرار بگیرد. | BPJWSSecAnyCpu.dll |

اولين گام براي راه‌اندازي، انتخاب كتابخانهPS مي‌باشد كه سازمان بايستي ضمن انتخاب درست فايل‌هاي مورد نياز خود بسته به مشخصات پلت فرم سرویس‌دهنده‌ی خود، فرايند ساخت نسخه و استقرار آن را به‌درستي انجام دهد.

در این مدل تعامل PS API وضعيت Client نسبت به زیرساخت HIX را دارد. وضعيت Client به معني اينكه نرم‌افزار تأمین‌کننده وب سرويس سوئيچ را فراخواني نموده و بايستي الزامات امنيتي سوئيچ و استفاده از كتابخانه PS را مطابق با اين دستورالعمل لحاظ نمايد.

* + 1. ثبت نام هویت سازمان و نرم‌افزار مبداء فراخوانی کننده

سازمان درخواست‌کننده‌ی اتصال باید درخواست اتصال به سامانه را طی تشریفات تفاهم‌نامه تحویل دهد. درخواست مذکور شامل اطلاعات هویتی و مختصات فنی سازمان می‌باشد که براساس آن سرفصل هویتی در سامانه‌ی مديريت سوئيچ تعریف می‌گردد.

سازمان درخواست‌کننده‌ی اتصال، بعد از انتخاب سرویس‌دهنده‌ی خود، برای ثبت آن، شناسه‌ی واحد آن را در درخواست خود درج و يا به كارشناسان واحد امنيت ابلاغ نمايد. برای استخراج شناسه‌ی فوق كه منطبق با سخت‌افزار سرویس‌دهنده است بايستي از كتابخانه PS استفاده گردد. فرايند به‌كارگيري كتابخانه شامل دو مرحله كلي مي‌باشد:

الف ) مرحله‌ی راه‌اندازي، يكپارچه‌سازي و استقرار (Initialization Step)

ب ) مرحله‌ی فراخواني وب‌سرويس‌هاي سوئيچ زیرساخت HIX (Usage Step)

در مرحله‌ی راه‌اندازي، تأمين‌كننده بايستي براساس جدول فايل‌ها و منطبق با پلت فرم سرویس‌دهنده، كتابخانه‌هاي مطابق با نرم‌افزار خود را انتخاب و فرايند يكپارچه‌سازي نرم‌افزار با كتابخانه‌هاي PS را اجرا نمايد.

لازم به ذكر است كه در زمان به‌كارگيري كتابخانه‌ی PS API براي فراخواني وب سرويس سوئيچ، نقش برنامه‌ی تأمین‌کننده، Client بوده و نمونه كدهاي مربوط به آن قابل بهره‌برداري مي‌باشد.

* + 1. صدور شناسه و سرفصل امنیتی برای متعاملین سوئيچ

هویت دیجیتال صادره برای مخاطبین سوئيچ در سه سطح امنیتیِ توکن نرم، توکن سخت و گذرواژه‌ی پویا ارائه می‌گردد و بنا به بضاعت، زیرساخت‌ها، الزامات، سیاست‌ها و مختصات فنی سازمان‌ها متفاوت و منعطف می‌باشد.

بعد از تأیید درخواست اتصال سازمان توسط سوئيچ، اعتبارنامه اتصال به وب‌سرویس در اختیار سازمان مربوطه قرار خواهد داد. گام‌هاي يكپارچه‌سازي با كتابخانه توابع:

1. قرار دادن dll يا so فايل مربوطه در برنامه‌ی تأمین‌کننده منطبق با پلت‌فرم برنامه‌ی مذکور با بارگذاري اين كتابخانه توابع زير در دسترس خواهد بود.

**public class** JniBPJWSSec  
{  
 **public native** String getVer();  
 **public native** String getID();  
 **public native** String passGen2(String rand1,String param1);  
 **public native boolean** validate2(String \_lock,String \_id,String pgen2,String rand1,String param1);  
}

1. استقرار برنامه‌ی نمونه بر روي سرویس‌دهنده‌ی نرم‌افزار تأمین‌کننده
2. فراخواني متد getID و به‌دست آوردن شناسه‌ی واحد سرویس‌دهنده براساس سرفصل‌هاي امنيتي
3. ارسال شناسه به واحد امنيت
4. اخذ نام كاربري یا شناسه‌ی سازمان متعامل (SenderId)

وب سرويس سوئيچ داراي سه پارامتر خاص براي امنيت مي‌باشد:

SenderId: نام کاربری كه در بخش قبل نحوه‌ی بدست آوردن آن بيان گرديد.

PlainId: شناسه‌ی آشکار كه به‌عنوان يك عدد تصادفي در هر بار فراخواني يا تراكنش با وب سرويس سوئيچ توسط نرم‌افزار تأمین کننده بايستي توليد شود. اين عدد تصادفي الزاماً بايد بصورت تصاعدي باشد چرا كه در غير اين صورت وب سرويس سوئيچ با خطای عدم معتبر بودن شناسه‌ی آشکار پاسخ خواهد داد.

(براي توليد شناسه آشكار مي‌توان از فرمت ترکیبی تاریخ و زمان به شکل yyyyMMddHHmmssfff در زبان‌هاي برنامه‌نويسي استفاده نمود كه داراي خصوصيات ذكر شده بالا نيز مي‌باشد).

EncryptedId: شناسه‌ی امن كه توسط متد passGen2 با پارامترهاي شناسه‌ی آشكار و نام كاربري توليد مي‌گردد.

String passGen2(String rand1,String param1);

rand1: شناسه آشكار

param1 : نام كاربري (PartyID)

* 1. نمونه کد فراخواني کانال دسترسی زیرساخت HIX به صورت همگام
     1. نمونه کد در .NET

نمونه کد برای فراخوانی متدهای REST در .NET :

using BoxBPJWSSec;

.

.

.

AutoLoad3264.path = AppDomain.CurrentDomain.BaseDirectory;

AutoLoad3264.Register("LZ4mm");

AutoLoad3264.Register("LZ4cc");

//Call A restful Service by Post Method

//Change sapaad-server with real sapaad service provider url

string senderId = "*sender\_id*";

string receiverId = "*receiver\_id*";

string url = "http://<SAPAAD\_ADDRESS>:8888/dispatch";

string plainId = DateTime.Now.ToString("yyyyMMddHHmmssfff");

String encryptId = LZ4Codec.passGen2(plainId, senderId);

//this values should be received from Sapaad and You get them

//in your web application and save them to reuse in calling our web services

var reqData = new

{

payload = new { PAYLOAD\_CLASS},

//fill payload variable with with parmeters that you want to send to target

payloadClass = "*payload\_class\_name*",

senderId= senderId,

receiverId= receiverId,

service= "*service\_name*", //fill with destination service name

action= "*action\_name*", //fill with destination action name

version= "*version*", //fill with destination service version number

timeout= "*time\_out*", //time for waiting for get service

plainId= plainId,

encryptedId= encryptedId

};

//Make Request

WebRequest request = WebRequest.Create(url);

request.Method = "POST";

request.ContentType = "application/json";

//serialize request in JSON by javaScriptSreializer class and get the string value

JavaScriptSerializer serializer = new JavaScriptSerializer();

string s = serializer.Serialize(reqData);

//Convert the string to utf-8 byte[]

byte[] byteArray = System.Text.UTF8Encoding.Unicode.GetBytes(s);

request.ContentLength = byteArray.Length;

//send Request and Get Response

Stream reqStream = request.GetRequestStream();

reqStream.Write(byteArray, 0, byteArray.Length);

reqStream.Close();

WebResponse response = request.GetResponse();

Stream resStream = response.GetResponseStream();

StreamReader reader = new StreamReader(resStream);

// the response is in Json format which includes payload result or Exception

string resString = reader.ReadToEnd();

reader.Close();

* + 1. نمونه کد درJAVA

نمونه کد برای فراخوانی متدهای REST در JAVA:

**public void** callPost(){

JniBPJWSSec \_j1=**new** JniBPJWSSec();

**public static** String *SERVER* =**"**http://*sapaad\_address*:8888/dispatch";  
 **public static** String *payloadClass* =**"***payload\_class\_name*";  
 **public static** String *senderId*="*sender\_id*";  
 **public static** String *receiverId* ="*receiver\_Id*";  
 **public static** String *serviceName*="*service\_name*";  
 **public static** String *actionName*="*action\_name*";  
 **public static** String *version*="*version*";  
 **public static** String *timeout*="*time\_out*";

String plainId = **new** SimpleDateFormat(**"yyyyMMddHHmmssS"**).format(**new** Date());  
 String encryptId= \_j1.passGen2(plainId.toString(), *senderId*);

RestTemplate rest = **new** RestTemplate();  
 Map<String, Object> params = **new** HashMap<String, Object>();  
 Map<String, Object> payloads = **new** HashMap<String, Object>();  
  
 payloads.put(**"***var\_name***"** , <VAR\_VALUE>);

// fill variable map value for send as payload to target service

params.put(**"payload"**, payloads);  
 params.put(**"payloadClass"**, *payloadClass*);  
 params.put(**"senderId"**, *senderId*);  
 params.put(**"receiverId"**, *receiverId*);  
 params.put(**"service"**, *serviceName*);  
 params.put(**"action"**, *actionName*);  
 params.put(**"version"**, *version*);  
 params.put(**"timeout"**, *timeout*);  
 params.put(**"plainId"**, *plainId*);  
 params.put(**"encryptedId"**, *encryptId*);  
  
 Map response = rest.postForObject(SERVER, params, Map.**class**);  
 System.***out***.println(response.toString());

}

* 1. سرویس‌های زیر ساخت

به منظور تسهیل تعامل سامانه‌های نرم افزاری بایکدیگر، در این نسخه چند سرویس ارزشمند جهت ارائه به سامانه‌های نرم‌افزاری متعامل ارائه گردیده است.

* + 1. دریافت لیست تفاهم‌نامه‌های مربوط به سازمان

براساس این سرویس سازمان‌ها می‌توانند توسط سامانه‌های نرم‌افزاری خود لیست تفاهم‌نامه‌های مربوط به خود در حالت سرویس‌دهنده و سرویس‌گیرنده را دریافت نمایند.

| ردیف | نام پارامتر | نوع | اجباری؟ | توضیح |
| --- | --- | --- | --- | --- |
| 1 | senderId | String | اجباری | کد سازمان |
| 2 | receiverId | String | اجباری | مقدار ثابت ix-almighty |
| 3 | service | String | اجباری | مقدار ثابت repo-central |
| 4 | action | String | اجباری | مقدار ثابت loadPartyAgreements |
| 5 | version | String | اجباری | مقدار ثابت 1.0.0 |
| 6 | timeout | Number | اختیاری | حداکثر زمان انتظار برای دریافت سرویس |
| 7 | payload |  |  | حاوی یک پارامتر به نام partyId که در آن می‌بایست کد سازمان تکمیل گردد. این سرویس تفاهم‌نامه‌های دریافت سرویس برای این party را گزارش می‌کند. |
| 7 | payloadClass | String | اجباری | مقدار ثابت  ix.microserver.repo.messages.\_  agreement.LoadPartyAgreementsRequest |
| 8 | plainId | String | اجباری | شناسه آشکار |
| 9 | encryptedId | String | اجباری | شناسه امن |

فراخوانی سرویس کانال دسترسی به صورت همگام در قالب json بصورت زیر است:

##request:

{

"payload" : {

"partyId" : "کد سازمان مبداء"

},

"payloadClass" : "ix.common.message.HeartbeatRequest",

"senderId" : "کد سازمان مبداء",

"receiverId" : "ix-almighty ",

"service" : "repo-central",

"action" : "loadPartyAgreements",

"version" : "1.0.0",

"timeout" :"مدت زمان انتظار برای دریافت پاسخ" ,

"plainId" :" شناسه آشکار ",

"encryptedId" :"شناسه امن",

}

مقدار بازگشتی از فراخوانی سرویس کانال دسترسی به صورت جدول ذیل ارائه می‌گردد:

| ردیف | نام پارامتر | | | | نوع | اجباری؟ | توضیح |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | payload | | | |  | اجباری | Object حاوی تفاهم‌نامه‌ها |
|  | agreements | | | |  | اجباری | محتوی لیستی از تفاهم‌های سازمان به صورت object |
|  |  | number | | | String | اجباری | شماره تفاهم در زیر ساخت |
|  |  | referToNumber | | | String | اجباری | شماره تفاهم قبلی که تفاهم جدید در ادامه آن است |
|  |  | producer | | |  | اجباری | Object حاوی اطلاعات سازمان سرویس‌دهنده |
|  |  |  | mcId | |  | اجباری | Object حاوی اطلاعات سرویس و سازمان مربوطه |
|  |  |  |  | partyId | String | اجباری | کد سازمان سرویس‌دهنده |
|  |  |  |  | Service | String | اجباری | نام سرویس مورد تفاهم |
|  |  |  |  | Version | String | اجباری | نسخه‌ی سرویس |
|  |  |  | action | | String | اجباری | نام عملیاتی از سرویس که مورد تفاهم است |
|  |  | producerPartyDisplayName | | | String | اجباری | نام نمایشی سازمان ارائه‌دهنده‌ی سرویس |
|  |  | consumerPartyId | | | String | اجباری | کد سازمان سرویس‌گیرنده |
|  |  | consumerPartyDisplayName | | | String | اجباری | نام نمایشی سازمان سرویس‌گیرنده |
|  |  | startDate | | | String | اجباری | تاریخ شروع تفاهم |
|  |  | endDate | | | String | اختیاری | تاریخ پایان تفاهم |
|  |  | timeout | | | Number | اجباری | مقدار زمان تاخیر تفاهم شده در دریافت سرویس در عملیات همگام. در سرویس‌های ناهمگام مقدار آن صفر است |
|  |  | timeToAck | | | Number | اجباری | مقدار زمان تاخیر تفاهم شده در دریافت درخواست در عملیات ناهمگام. در عملیات همگام مقدار آن صفر است |
|  |  | timeToPerform | | | Number | اجباری | مقدار زمان تاخیر تفاهم شده در ارسال پاسخ در عملیات ناهمگام. در عملیات همگام مقدار آن صفر است |
|  |  | temporaryDeactivate | | | Boolean | اجباری | نشان‌دهنده‌‎‎ی این است که آیا این تفاهم موقتاً غیر فعالی است؟ |
|  |  | active | | | Boolean | اجباری | نشان‎دهنده‌ی فعال بودن یا نبودن تفاهم |
|  |  | timeUnit | | | Number | اجباری | واحد زمانی قابلیت استفاده از سرویس شامل:   * DAYS : به معنای روز * ... |
|  |  | Duration | | | Number | اجباری | طول مدت زمان قابلیت استفاده از سرویس براساس واحد timeUnit |
| 2 | type | | | | String | اختیاری | مقدار ثابت ORDINARY |
| 3 | payloadClass | | | | String | اجباری | مقدار ثابت  ix.microserver.repo.messages.agreement.\_  LoadPartyAgreementsResponse |
| 4 | conversationId | | | | Sring | اختیاری | Null |

مقادیر بازگشتی از سرویس کانال دسترسی به صورت همگام در قالب json به‌صورت زیر است:

##response

{

"payload" : {

"agreements" : [

{

"number": "شماره تفاهم نامه 1",

"referToNumber": null,

"producerPartyDisplayName":"سازمان من",

"consumerPartyDisplayName":"سازمان 1",

"producer": {

"mcId": {

"partyId": "کد سازمان من",

"service": "نام سرویس",

"version": "شماره نسخه سرویس"

},

"action": "نام عملیات مربوط به سرویس",

},

"consumerPartyId": "کد سازمان 1",

"startDate": "تاریخ شروع",

"endDate": null,

"timeOut": 30,

"timeToAck": 0,

"timeToPerform": 0,

"active": true,

"timeUnit": "DAYS",

"duration": 100,

"temporaryDeactivate": false

},

{

"number": "شماره تفاهم نامه 2",

"referToNumber": null,

"producerPartyDisplayName":"سازمان 2",

"consumerPartyDisplayName":"سازمان من",

"producer": {

"mcId": {

"partyId": "کد سازمان 2",

"service": "نام سرویس",

"version": "شماره نسخه سرویس"

},

"action": "نام عملیات مربوط به سرویس",

},

"consumerPartyId": "کد سازمان من",

"startDate": "تاریخ شروع",

"endDate": null,

"timeOut": 0,

"timeToAck": 30,

"timeToPerform": 180,

"active": true,

"timeUnit": "DAYS",

"duration": 200,

"temporaryDeactivate": false

}

]

},

"payloadClass" : "ix.microserver.repo.messages.agreement.LoadPartyAgreementsResponse",

"type" : "ORDINARY",

"conversationId" : null

}

در خروجی بالا سازمان من دارای دو تفاهم نامه است:

* در تفاهم‌نامه‌ی اول سرویس‌دهنده است؛ سازمان من در این تفاهم به مدت 100 روز به سازمان 1 سرویس ارائه   
  می‌کند و سرویس فوق به‌صورت همگام می‌باشد.
* در تفاهم‌نامه‌ی دوم سرویس‌گیرنده است. سازمان من در این تفاهم به مدت 200 روز از سازمان 2 سرویس دریافت   
  می‌کند و این سرویس به صورت ناهمگام است.
  + 1. آگاهی از وضعیت سرویس در سازمان مقصد

سازمان های سرویس‌گیرنده می‌توانند توسط سامانه‌های نرم‌افزاری خود به واسطه‌ی این سرویس از وضعیت سرویسِ سازمان مقصد مطلع گردند.

| ردیف | نام پارامتر | نوع | اجباری؟ | توضیح |
| --- | --- | --- | --- | --- |
| 1 | senderId | String | اجباری | کد سازمان مبداء |
| 2 | receiverId | String | اجباری | کد سازمان مقصد |
| 3 | service | String | اجباری | نام سرویس در سازمان مقصد |
| 4 | action | String | اجباری | نام عملیات مربوط به سرویس در سازمان مقصد |
| 5 | version | String | اجباری | شماره نسخه مربوط به سرویس |
| 6 | timeout | Number | اختیاری | حداکثر زمان انتظار برای دریافت سرویس |
| 7 | payloadClass | String | اجباری | مقدار ثابت  ix.common.message.HeartbeatRequest |
| 8 | plainId | String | اجباری | شناسه آشکار |
| 9 | encryptedId | String | اجباری | شناسه امن |

فراخوانی سرویس کانال دسترسی به صورت همگام در قالب json به‌صورت زیر است:

##request:

{

"payload" : null,

"payloadClass" : "ix.common.message.HeartbeatRequest",

"senderId" : "کد سازمان مبداء",

"receiverId" : "کد سازمان مقصد",

"service" : "نام سرویس در سازمان مقصد",

"action" : "نام عملیات در سرویس سازمان مقصد",

"version" : "نسخه مربوط به سرویس سازمان مقصد",

"timeout" :"مدت زمان انتظار برای دریافت پاسخ" ,

"plainId" :" شناسه آشکار ",

"encryptedId" :"شناسه امن",

}

مقدار بازگشتی از فراخوانی سرویس کانال دسترسی به صورت جدول ذیل ارائه می‌گردد:

| ردیف | نام پارامتر | | نوع | اجباری؟ | توضیح |
| --- | --- | --- | --- | --- | --- |
| 1 | payload | |  |  |  |
|  | status | | Boolean | اجباری | دارای مقادیر ذیل:   * true : به معنای این است که سرویس‌دهنده‌هایی فعال است. * false : به معنای این است که کانال تحویل زیرساخت فعال است لیکن سرویس‌دهنده‌ی نهایی غیرفعال است. |
|  | microServerId | |  |  | مشخصات سرویس نهایی |
|  |  | partyId | String | اجباری | کد سازمان مقصد |
|  |  | service | String | اجباری | نام سرویس |
|  |  | version | String | اجباری | شماره نسخه‌ی مربوط به سرویس |
| 2 | type | | String | اختیاری | مقدار ثابت ORDINARY |
| 3 | payloadClass | | String | اجباری | مقدار ثابت  ix.common.message.HeartbeatRequest |
| 4 | conversationId | | Sring | اختیاری | null |

مقادیر بازگشتی از سرویس کانال دسترسی به صورت همگام در قالب json به‌صورت زیر است:

##response

{

"payload" : {

"status" : true/false,

"microServerId" : {

"partyId" : "کد سازمان مقصد",

"service" : "نام سرویس",

"version" : "شماره نسخه سرویس"

}

},

"payloadClass" : "ix.common.message.HeartbeatResponse",

"type" : "ORDINARY",

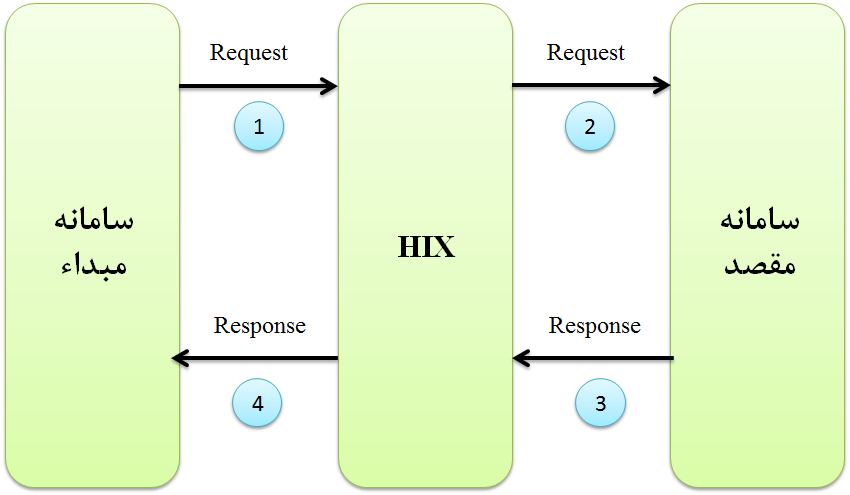
"conversationId" : null

}

\* در صورت بروز خطای زمانی (timeout) مفهوم این است که کانال تحویل سازمان مقصد غیرفعال است. به منظور آگاهی از سایر خطاهای احتمالی به بخش توصیف خطاهای فراخوانی در همین سند مراجعه شود.

1. پیوست‌ها
   1. پیوست 1: توصیف مدل ارتباطی همگام

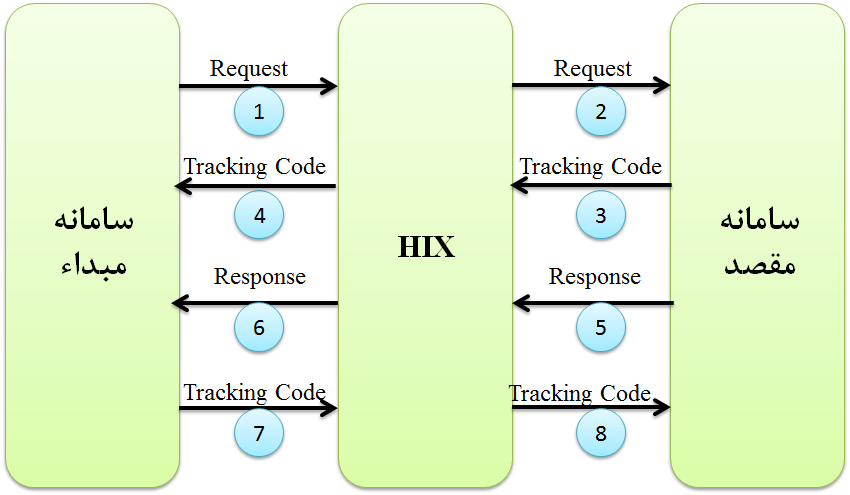
دراین مدل در ابتدا سازمان سرویس‌گیرنده (که در شکل با سازمان مبداء مشخص گردیده است) درخواست خود را به زیرساخت اعلان می‌نماید، سپس زیرساخت درخواست وی را برای سازمان سرویس‌دهنده ارسال خواهد کرد. از آنجایی که ارتباط به‌صورت همگام صورت می‌پذیرد سرویس‌دهنده ملزم به ارائه‌ی پاسخ در همان لحظه می‌باشد. بدین ترتیب سازمان سرویس‌دهنده پاسخی را آماده و برای زیرساخت ارسال می‌نماید که به‌طبع آن زیرساخت نیز پاسخ دریافتی را برای سازمان درخواست‌کننده مهیا می‌سازد. بدیهی است که در طول این رویه‌ی رفت و برگشت، سازمان سرویس‌گیرنده منتظر دریافت پاسخ از سازمان سرویس‌دهنده بوده و تا دریافت پاسخ بلوکه خواهد شد. در عین حال سازمان مبداء برای درخواستی که مطرح نموده است زمانی را به عنوان شاخص جهت دریافت پاسخ مشخص خواهد نمود؛ چنانچه در زمان مشخص شده سامانه‌ی مقصد و به‌تبع آن زیرساخت پاسخ را در اختیار سامانه‌ی مبداء قرار دهد رویه به‌صورت کامل و بی‌نقص پیش رفته است. حال اگر در زمان تعیین شده پاسخی در اختیار سامانه‌ی مبداء قرار نگیرد، سامانه‌ی مبداء می‌تواند مجدداً درخواست خود را مطرح نماید. در چنین مواردی چند حالت در ایجاد عدم پاسخ‌گویی ممکن است نقش داشته باشند: 1) اشکالی در زیرساخت وجود داشته باشد. 2) اشکالی در سامانه‌ی مقصد باشد. 3) سامانه‌ی مقصد زمان بیشتری جهت پاسخ‌گویی نیاز داشته باشد.



شکل شماره 15 : فرآیند درخواست و دریافت پاسخ در مدل ارتباطی همگام

* 1. پیوست 2: توصیف مدل ارتباطی ناهمگام

مدل ارتباطی ناهمگام خود دربردارنده‌ی دو شکل ارتباطی کلی است که در ادامه به بررسی جزئیات در هریک از دو مدل می‌پردازیم. مدل ارتباطی ناهمگام اول دربردارنده‌ی دو رویه‌ی ارتباطی همگام است، بدین ترتیب که در ابتدا سازمان سرویس‌گیرنده (مبداء) درخواست خود را به زیرساخت اعلان می‌نماید و زیرساخت درخواستِ صادره را در اختیار سازمان سرویس‌دهنده قرار می‌دهد. با دریافت پیام توسط سازمان مقصد، کد رهگیری مبتنی بر دریافت پیام توسط وی به سامانه‌ی مبداء ارسال می‌شود. بدیهی است با دریافت کد رهگیری توسط سامانه‌ی مبداء دو نکته‌ی حائزاهمیت وجود دارد: اول اینکه به سامانه‌ مبداء اعلان می‌شود که پیام به دست سامانه‌ی مقصد رسیده است و از ارسال مجدد پیام خودداری نماید چراکه سامانه‌ی مقصد در حال بررسی و آماده‌سازی پاسخ است. دوم اینکه سامانه‌ی مبداء با دریافت کد رهگیری به باقی کارهای خود می‌رسد و منتظر دریافت پاسخ نمی‌ماند چراکه به‌محض آماده شدن پاسخ توسط سامانه‌ی مقصد، پیام پاسخ برای سامانه‌ی مبداء ارسال خواهد شد بدین ترتیب از بلوکه شدن سامانه‌ی مبداء جلوگیری خواهد شد. بنابراین همان‌طور که به‌سادگی می‌توان استنتاج نمود رویه‌ی ارسال درخواست و دریافت کد رهگیری رویه‌ای همگام می‌باشد. در فاز دوم با آماده شدن پاسخ در سامانه‌ی مقصد این پیام برای سامانه‌ی مبداء ارسال خواهد شد و در مقابل سامانه‌ی مبداء با دریافت آن پیغامی مبتنی بر دریافت صحیحِ پاسخ ارسال می‌نماید. فاز دوم نیز همانند فاز اول دارای رویه‌ای همگام می‌باشد. در نتیجه مدل ارتباطیِ ناهمگام فوق دارای دو رویه‌ی همگام است که سبب می‌شود سامانه‌ی مبداء در درجه‌ی اول از دریافت درخواست خود توسط سامانه‌ی مقصد اطمینان حاصل نماید، دوم اینکه سامانه‌ی مبداء پس از اعلان درخواست منتظر دریافت پاسخ نبوده و به کارهای خود رسیدگی کند.

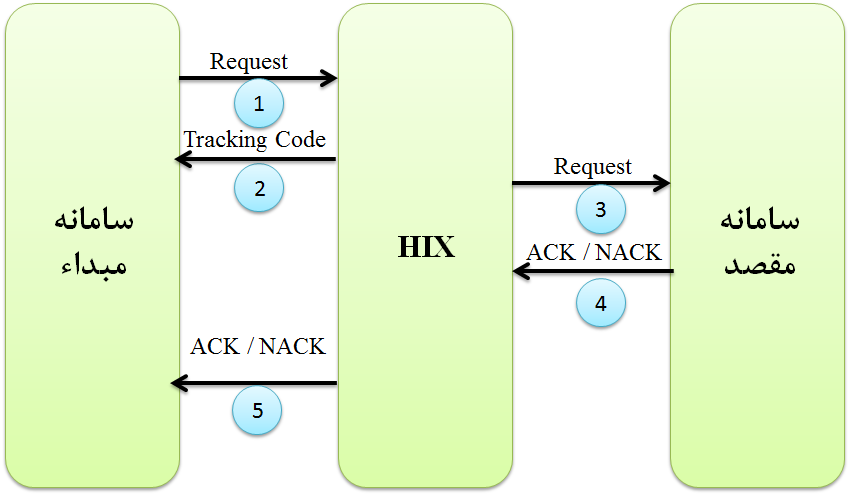


شکل شماره 16 : فرآیند درخواست و دریافت پاسخ در مدل ارتباطی ناهمگام (ترکیبی از دو مدل همگام)

در نوع دوم از مدل ارتباطی ناهمگام، ترکیبی از رویه‌های همگام و ناهمگام را خواهیم داشت: بدین ترتیب که درابتدا سامانه‌ی مبداء درخواست خود را به زیرساخت ارسال و زیرساخت خود کد رهگیریِ مشخصی به‌منظور اعلام دریافت صحیحِ درخواست به سامانه‌ی مبداء خواهد داد. سامانه‌ی مبداء به‌محض دریافت این کد رهگیری نگرانی‌ایی از بابت ارسال پیام به سامانه‌ی مقصد نخواهد داشت و این مهم را برعهده‌ی زیرساخت نهاده است و بدین ترتیب منتظر نمانده و تا اعلان پاسخ به کارهای خود رسیدگی می‌نماید.

در ادامه زیرساخت موظف است درخواست دریافتی از سامانه‌ی مبداء را به سامانه‌ی مقصد ارسال نماید. بدین ترتیب به‌جای انتظارِ سامانه‌ی مبداء، زیرساخت منتظر دریافت پاسخ دریافت از سامانه‌ی مقصد خواهد بود و انتظار به سمت زیرساخت منتقل می‌شود. زیرساخت پس از مدت زمان معینی پاسخ دریافت درخواست را از سامانه‌ی مقصد دریافت و آن را به سامانه‌ی مبداء منتقل می‌نماید.

در چنین مدل ارتباطیِ ناهمگامی، در فاز اول حالت همگام را داریم چراکه سامانه‌ی مبداء با ارسال درخواست همان لحظه منتظر دریافت کد رهگیری است و با دریافت آن از حالت بلوکه خارج و به فعالیت خود رسیدگی می‌نماید. در عین حال در فاز دوم رویه‌ی ناهمگام را خواهیم داشت چراکه سامانه‌ی مبداء تا دریافت تأییدیه‌ی دریافت / عدم دریافت پیام از جانب سامانه‌ی مقصد منتظر نمانده و سپس پاسخ دریافت/عدم دریافت پیام به سامانه‌ی مبداء ارسال می‌گردد.



شکل شماره 17 : فرآیند درخواست و دریافت پاسخ در مدل ارتباطی ناهمگام (ترکیبی از رویه‌ی همگام و ناهمگام)

برای هریک از عملیات زمانی مشخص خواهد شد که اگر چنانچه در زمان مشخص شده، عمل مقرر صورت نپذیرد، پیغامی متناسب با آن ارائه نماید. پاسخ به درخواست نیز روندی مشابه‌ی ارسال درخواست در جهت مخالف خواهد داشت که برای جلوگیری از شلوغی شکل در آن نشان داده نشده است.