

## Incident management

### 1- On-Call Engineer

#### شرح دقیق نقش و مسئولیت‌های On-Call Engineer در مدیریت حوادث

در سامانه‌های حساس و حیاتی نظیر کارگزاری بورس آنلاین، پایداری و دسترس‌پذیری سیستم اهمیت زیادی دارد. بخشی از الزامات تضمین این پایداری، داشتن فرآیندی شفاف برای مدیریت حوادث (Incident Management) است.

در همین راستا، نقش On-Call Engineer یا «اپراتور مسئول پاسخگویی برخط» تعریف و وظایف آن به صورت زیر شفاف‌سازی می‌شود:

#### تعريف On-Call Engineer

On-Call Engineer فردی از اعضای تیم پشتیبانی یا فنی است که در هر بازه زمانی مشخص، مسئولیت نظارت، شناسایی و واکنش سریع به رخدادهای غیرمنتظره، خطاهای مشکلات مهمنامه را بر عهده دارد. این نقش به صورت چرخشی (شیفتی) میان اعضای تیم فنی توزیع می‌شود تا همواره یک شخص آماده پاسخگویی وجود داشته باشد.

#### وظایف کلیدی On-Call Engineer

پایش مستمر سامانه: تحت نظر داشتن سلامت کلی سرویس‌ها، بررسی سامانه‌های مانیتورینگ و دریافت هشدارها (Alerts)، چه به صورت اتوماتیک و چه دستی.

پاسخگویی سریع به رخدادها: دریافت و تحلیل رخدادهای رخداده (اعم از خطاهای سیستمی، کندی سرویس، افزایش تاخیر یا افت کیفیت) و اقدام فوری برای کاهش تأثیر یا رفع مشکل، بر اساس رویه‌های سازمانی و سطح اهمیت خطا.

هماهنگی با تیم توسعه و سایر مسئولین: در صورت رفع نشدن مشکل، اطلاع‌رسانی به سایر اعضای تیم، مدیریت تصاعد مشکل (escalation) و همکاری برای حل ریشه‌ای (Root Cause Analysis).

مستندسازی فعالیت‌ها: ثبت رخدادها، پاسخ‌ها و راهکارهای اعمال شده جهت مرور، یادگیری و بهبود فرآیند در آینده.

پس از رفع حادثه، گزارشی شامل اطلاعات زیر تهیه می‌شود:

شرح حادثه

دلیل وقوع

اقدامات انجام شده

پیشنهادات برای جلوگیری از حادثه مشابه

ابزار	نقش
Python	زبان اصلی توسعه سرویس‌ها
FastAPI	چارچوب طراحی RESTful برای پیاده‌سازی سرویس ثبت حادثه
BlockchainAdapter	ماژولی برای اتصال به بلاکچین و ثبت اطلاعات حادثه
Prometheus	ابزار مانیتورینگ و تشخیص مشکلات سیستم
Alertmanager	ابزار هشداردهی برای اطلاع‌رسانی سریع به مهندس آنکال

گزارش دهی به مدیریت: ارائه گزارش شفاف در پایان هر شیفت یا رخداد مهم، شامل لیست مشکلات، نحوه پاسخگویی و پیشنهادات لازم برای پیشگیری‌های بعدی.

معیارهای تعیین و شیفت‌بندی

برنامه‌ریزی شیفت‌های On-Call بر اساس تعداد اعضاء، مهارت فنی، حجم تراکنش‌ها و ساعت بحرانی انجام می‌شود.

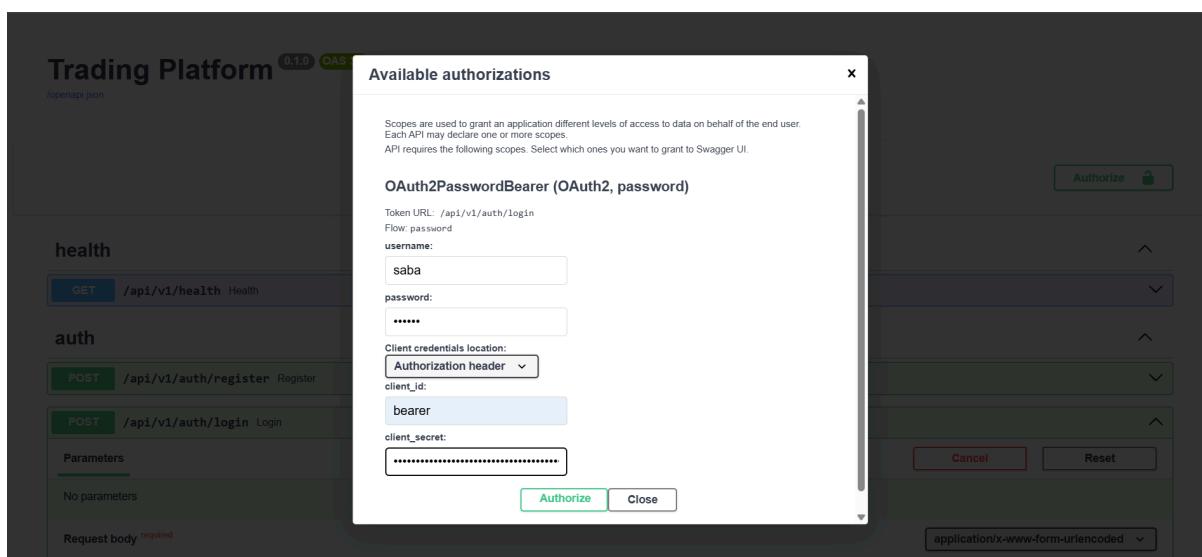
لیست اعضای مسئول و زمان‌بندی شیفت‌ها باید به صورت مستند و در دسترس کل تیم باشد. در صورت بروز تغییر در تقسیم وظایف، باید فوراً به همه اعضاء اطلاع‌رسانی گردد.

اهداف ایجاد نقش

افزایش سرعت شناسایی و حل مشکلات به منظور به حداقل رساندن زمان Down بودن سرویس

تصمیم ارائه سرویس پایدار و با کیفیت به کاربران نهایی  
به حداقل رساندن ریسک‌های ناشی از رخدادهای پیش‌بینی‌نشده و کاهش اثر آن‌ها بر کسب‌وکار و اعتبار  
سامان

blockchain owner بر تخصیص کلید  
در swagger این تغییر را تست کردم:  
ابتدا با یک کاربر رجیستر و لوگین می‌کنم و توکن داده شده را برای احراز هویت استفاده می‌کنم



اینجا توanstم owner قرارداد را ببینم:

Request URL  
`http://localhost:8000/api/v1/admin/blockchain/owner`

Server response

Code	Details
200	<p>Response body</p> <pre>{ "owner": "0x71562b71999873D85b286dF957af199Ec94617F7" }</pre> <p>Download</p> <p>Response headers</p> <pre>content-length: 54 content-type: application/json date: Wed, 11 Feb 2026 11:16:32 GMT server: unicorn</pre>

Responses

Code	Description	Links
------	-------------	-------

در این قسمت هم سطح دسترسی بررسی شده که درسته و admin هست:

**address \*** required  
string  
(path)

0x71562b71999873DB5b286dF957af199Ec\$

**Execute**      **Clear**

---

**Responses**

**Curl**

```
curl -X 'GET' \
  'http://localhost:8000/api/v1/admin/blockchain/access-level/0x71562b71999873DB5b286dF957af199Ec94617F' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInRcIjIkpXVCJ9.eyJzdWIiOiIyIiwidXNwIjoxNzcwODAwfzcyfQ.X4uooUgLnSH2IDKtmIWG_-qHTtvbmAtGBr6kQS6eqiQ'
```

**Request URL**

```
http://localhost:8000/api/v1/admin/blockchain/access-level/0x71562b71999873DB5b286dF957af199Ec94617F
```

**Server response**

Code	Details
200	<b>Response body</b> { "address": "0x71562b71999873DB5b286dF957af199Ec94617F", "level": 3, "level_name": "ADMIN" }  <b>Response headers</b>

مثلا اگر یک آدرس پر ریط میداریم:

address \* required  
string  
(path)

0x00

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8000/api/v1/admin/blockchain/access-level/0x0000000000000000000000000000000000000000000000000000000000000001' \
-H 'accept: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cIj6IkpxVCJ9.eyJzdWIiOiIyIiwIXHwIjoxNzcxODEwMzcyfQ.X4uoUglnSH2IDktmJMV6_-qHTtvbmAtGBr9kQS6eqi'
```

Request URL

```
http://localhost:8000/api/v1/admin/blockchain/access-level/0x000000000000000000000000000000000000000000000000000000000000001
```

Server response

Code	Details
200	Response body
	<pre>{ "address": "0x00", "level": 0, "level_name": "NONE" }</pre>

Download

حالا قابلیت تعیین سطح دسترسی را، یک کاربر را تست ممکنیم:

Execute      Clear

### Responses

```
curl -X 'POST' \
  'http://localhost:8000/api/v1/admin/blockchain/set-access' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer eyJhbGkoiOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWJpZGVyIiwidXNlcnRzIjoxNzcsX4uud0glnSH2IDKtmDNG_--qHTtvbmAtG8r9kQS6q1Q' \
  -H 'Content-type: application/json' \
  -d '{
    "user_address": "0x123456789012345678901234567890",
    "level": 2
}'
```

Request URL  
<http://localhost:8000/api/v1/admin/blockchain/set-access>

Server response

Code	Details
200	Response body <pre>{   "success": true,   "tx_hash": "0xc32356dff3c3640affc268ffca7c35e7baaf124cd4aaa15cac0ce13f3be2ad843",   "message": "Access level set to 2" }</pre> <p>Download</p>
	Response headers <pre>content-length: 129 content-type: application/json date: Wed, 11 Feb 2020 11:31:37 GMT server: unicorn</pre>

که درست کار میکند.

### خلاصه‌ی سطوح دسترسی

توانایی‌ها	نقش	سطح
تخصیص/حذف همه نقش‌ها توقف/راه‌اندازی مجدد سیستم <b>recorders</b> مجوزدهی به <b>incidents</b> حل کردن انتقال مالکیت	مالک <b>Contract</b>	<b>OWNER</b>
توقف اضطراری سیستم - حل کردن <b>incidents</b>	مدیر اضطراری	<b>EMERGENCY_ADMIN</b>
<b>incidents</b> گزارش	اپراتور	<b>OPERATOR</b>
<b>trades</b> ثبت	معامله‌گر	<b>TRADER</b>
<b>trades</b> ثبت	ثبت‌کننده مجاز	<b>Authorized Recorder</b>

روش‌های دیگری که میشد استفاده کرد:

- 1- استفاده از الگوی کنترل مبتنی بر چند امضا (**Multi-Signature Wallet**)  
به جای اینکه Owner یک آدرس باشد، مالکیت و کنترل به چندین آدرس واگذار می‌شود. اجرای عملکردهای حساس (مثل تغییر تنظیمات، آپدیت قرارداد و ...) تنها با تایید اکثریت امضاکنندگان ممکن است (مثلاً  $n$  از  $m$ ).  
ابزارهای مشهور مثل Gnosis Safe این روش را بسیار رایج و عملی کرده‌اند.

## **2- پیاده‌سازی (DAO (Decentralized Autonomous Organization**

سطح دسترسی مالک به مجموعه‌ای از سهامداران یا اعضای سازمان غیرمتمرکز منتقل می‌شود. تصمیمات کلیدی و تغییرات با رای‌گیری (governance) انجام می‌شود (مثلاً با توکن‌های gov یا (voting power).

قراردادهای معروف DAO وجود دارند که امکان ارتقای سیستم را به صورت غیرمتمرکز فراهم می‌کنند (مانند OpenZeppelin Governor).

## **3- استفاده از قراردادهای قابل ارتقا با مدیریت جمعی (Upgradeable Contracts with Collective Control**

به جای owner، یک Governor یا Proxy Admin انتخاب می‌شود که خودش تحت مدیریت گروه DAO است.

دسترسی ارتقا و مدیریت به مجموعه‌ای از کاربران یا نهادها تعلق می‌گیرد.

## **4- حذف کامل مالکیت (Renounce Ownership**

پس از راه‌اندازی، مالکیت به طور کامل حذف و هیچ آدرس خاصی کنترل ندارد. توجه شود که بدون امکان مدیریت، ارتقای سیستم سخت‌تر خواهد شد.

## **5- Timelock Controller (تاخیر زمانی بر عملیات حساس)**

اجرای عملیات مهم قرارداد (مثلاً ارتقاء یا تغییر مالکیت) را با تاخیر زمانی (مثلاً ۲ روز) همراه می‌کند.

در این مدت همه می‌توانند تغییرات قریب الوقوع را ببینند و در صورت لزوم واکنش نشان دهند. معمولاً همراه با multisig/DAO استفاده می‌شود و امنیت و شفافیت را افزایش می‌دهد.