АКЦИОНЕРНОЕ ОБЩЕСТВО
«ЦЕНТР ОТКРЫТЫХ СИСТЕМ
И ВЫСОКИХ ТЕХНОЛОГИЙ»

Программный комплекс ситуационного анализа (ядро операционного центра) COSOC

РУКОВОДСТВО АДМИНИСТРАТОРА

Москва, 2018

Содержание

Определения, обозначения и сокращения 3
1 Введение 5
1.1 Назначение программного комплекса 5
1.2 Сведения о программных и технических средствах . . . 5
1.2.1 Сведения о программных средствах 5
1.2.2 Сведения о технических средствах 6
1.2.3 Требование к персоналу (системному администратору) 7
2 Структура программного комплекса 8
2.1 Сведения о структуре программного комплекса 8
2.2 Состав компонентной структуры 10
2.2.1 Состав и описание компонентов общего назначения 10
2.2.2 Состав и описание компонентов специального программного обеспечения 11
2.3 Взаимодействие между компонентами и внешними системами 12
3 Обязанности администратора и связанные с ними операции . . 13
3.1 Развертывание программного комплекса COSOC 13
3.1.1 Установка Python 2.7 13
3.1.2 Установка COSOC 14
3.2 Использование REST API 14
3.2.1 Проверка работоспособности API 14
3.2.2 Отправка CAP-сообщений 14
3.2.3 Выборка CAP-сообщений 15
3.2.4 Получение оригинала CAP-сообщения 18
3.2.5 Получение сообщения в формате json 20
4 Сообщения об ошибках 23

Определения, обозначения и сокращения

API — (от англ. Application Programming Interface) — программный интерфейс приложения — набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением для использования во внешних программных продуктах

CAP — (от англ. Common Alerting Protocol) - протокол общего оповещения - стандарт OASIS, утвержденный FEMA (Federal Emergency Management Agency, определяющий формат xml-сообщения о событиях и угрозах

CSV — (CSV от англ. Comma-Separated Values — значения, разделённые запятыми) — текстовый формат, предназначенный для представления табличных данных

Dynamic HTML — набор средств, которые позволяют создавать более интерактивные Web-страницы без увеличения загрузки сервера

HTML — Язык гипертекстовой разметки документов (от англ. Hypertext Markup Language – "язык гипертекстовой разметки")

HTTP — Протокол прикладного уровня для передачи данных, используемый в Web (от англ. HyperText Transfer Protocol - «протокол передачи гипертекста»)

IP-адрес — Уникальный сетевой адрес узла в компьютерной сети, построенной по протоколу IP

JavaScript — Прототипно-ориентированный сценарный язык программирования. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам

J2EE — Java 2 Enterprise Edition, или Java Platform Enterprise Edition — набор спецификаций и соответствующей документации для языка Java, описывающие архитектуру серверной платформы для задач средних и крупных предприятий

JPEG (JPG) — JPEG - один из популярных графических форматов, применяемый для хранения фотоизображений и подобных им

изображений. Файлы, содержащие данные JPEG, обычно имеют расширения .jpg, .jfif, .jpe или .jpeg.

NGNIX — (от англ. engine x) — простой веб-сервер и почтовый прокси-сервер, не перегруженный лишними функциями, работающий на Unix-подобных операционных системах; применяется прежде всего для статических веб-сайтов и как прокси-сервера перед динамическими сайтами

PDF — Portable Document Format (PDF) — межплатформенный формат электронных документов, разработанный фирмой Adobe Systems

АС — Автоматизированная система

МЭДО — межведомственный электронный документооборот

НСИ — Нормативно–справочная информация

Интернет — Информационно-телекоммуникационная сеть Интернет

ИТ — Информационные технологии, информационно-технологический

Открытые данные — Информация, размещаемая ее обладателями в сети «Интернет» в формате, допускающем автоматизированную обработку без предварительных изменений человеком в целях повторного ее использования

ПО — Программное обеспечение

СМЭВ — Система межведомственного электронного взаимодействия

1 Введение

1.1 Назначение программного комплекса

Программный комплекс ситуационного анализа COSOC предназначен для обеспечения информационно-аналитической поддержки процессов сбора, хранения и обработки сообщений о событиях и угрозах, поступающих в форматированном виде от внешних источников информации.

Предметом автоматизации COSOC являются деловые процессы в том числе процессы информационного взаимодействия внутреннего и внешнего характера.

Программные комплекс не предназначен для использования в качестве самостоятельного решения, а используется как подсистема приёма и обработки сообщений в составе аппаратно-программных комплексов ситуационных или операционных центров, таких как, например, АПК "Безопасный город".

1.2 Сведения о программных и технических средствах

1.2.1 Сведения о программных средствах

Программный комплекс состоит из следующих компонентов:

1) модуль ситуационного анализа – обеспечивает прием, обработку и анализ специфицированных сообщений о событиях безопасности CAP формата;

2) сервер управления базами данных – обеспечивает централизованное хранение и управление данными, обрабатываемыми COSOC.

Все указанные выше компоненты должны быть физически запущены на одном сервере.

Доступ к данным осуществляется на основе клиент-серверной технологии с использованием REST API. Клиентская сторона приложения может быть реализована как web-приложение, работающее в браузере непосредственно на АРМ пользователя подсистемы. Серверная сторо-

на функционирует на специализированном программно-аппаратном комплексе.

Взаимодействие клиентской и серверной частей приложения осуществляется либо через локальную вычислительную сеть (ЛВС), либо через Интернет с использованием протоколов TCP/IP, HTTP(s), REST.

Количество рабочих станций, подключаемых к серверному компоненту, не регламентируется и ограничивается пропускной способностью каналов подключения и характеристиками серверного аппаратного обеспечения. Программный комплекс обеспечивает высокую масштабируемость.

Серверные компоненты программного комплекса функционируют под управлением операционной системы (ОС) RedHat Linux (CentOS) 6;

В качестве платформы для сбора и обработки информации о событиях используется нереляционная база банных Cassandra.

Комплекс разработан на базе технологии J2EE, что обеспечивает ряд существенных преимуществ при создании сложных промышленных программных комплексов, а именно – высокая производительность, кроссплатформенность, надежность, масштабируемость, гибкость.

В качестве программного web - сервера используется Apache Tomcat (контейнер сервлетов с открытым исходным кодом, разрабатываемый Apache Software Foundation).

1.2.2 Сведения о технических средствах

Программный комплекс функционирует на вычислительном комплексе, включающем:

Сервер с характеристиками не хуже чем XEON 2.8GHz 4CPUx4Core, 16Gb RAM, HDD 2x500Gb

1.2.3 Требование к персоналу (системному администратору)

Системный администратор должен иметь минимум среднее образование и знания на уровне администратора системы следующих программных компонент:

1) ОС RedHat (CentOS) 6;
2) Apache Tomcat 7;
3) Apache Cassandra;

В перечень задач, выполняемых системным администратором, входят:

— Задача поддержания работоспособности технических средств;

— Задача установки (инсталляции) и поддержания работоспособности общего программного обеспечения;

— Задача установки (инсталляции) и поддержания работоспособности программного комплекса.

2 Структура программного комплекса

2.1 Сведения о структуре программного комплекса

Программно-аппаратная структура программного комплекса показана на рисунке 2.1.

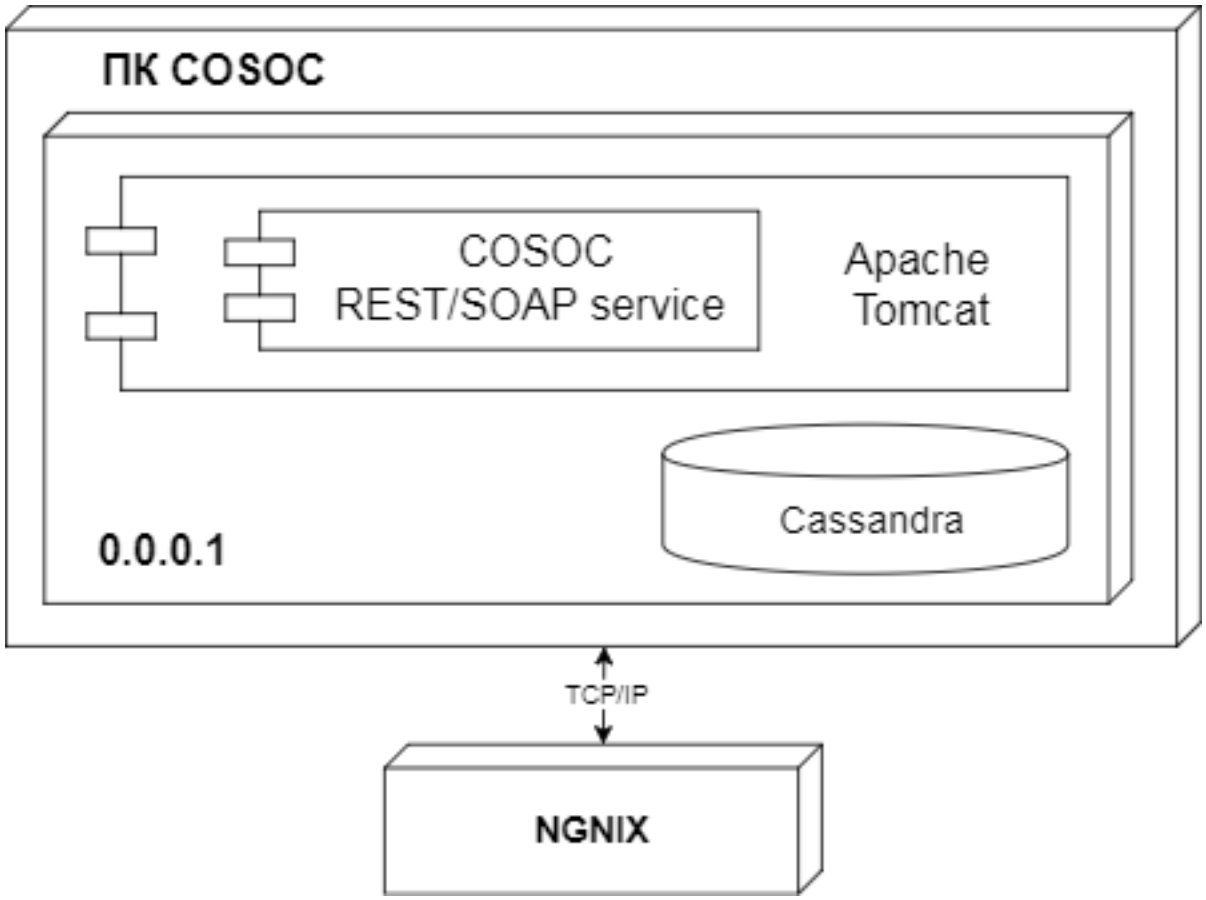

Рисунок 2.1 — Архитектура системы

В качестве основного канала получения сообщений используется Интернет, к которому COSOC подключается через какой-либо web-сервер. На рисунке 2.1 приведён web-сервер NGNIX, в качестве одного из возможных.

В таблице 2.1 приведены наименования компонент и программное обеспечение, на основе которого они реализованы.

Таблица 2.1 — Состав программных компонент

N п/п	**Наименование компонента**	**Используемое программное обеспечение**	**Функции компоненты**
1	Модуль ситуационного анализа	COSOC 1.1.3	Обеспечивает сбор, обработку и маршрутизацию информации о событиях и угрозах.
2	Сервер приложений	Apache Tomcat 7.0.26	Обеспечивает среду исполнения кодов программного модуля COSOC
2	Сервер нереляционной базы данных	Cassandra 2.2.10	Обеспечивает хранение всей информации и метаинформации компонентами программного комплекса за исключением файлов, хранящихся в файловой системе

2.2 Состав компонентной структуры

Программный комплекс строится на основе компонентов двух видов:

— компоненты общего назначения – программные компоненты с открытым кодом, обеспечивающие среду для построения специального программного обеспечения;

— специальные компоненты – специальное программное обеспечение (далее - СПО) - разрабатываемые программные компоненты, непосредственно реализующие функции программного комплекса.

2.2.1 Состав и описание компонентов общего назначения

В состав компонентов общего назначения входят следующие позиции:

1) операционная система (далее по тексту – ОС) – комплекс программных средств, выступающих в качестве интерфейса между аппаратными средствами и прочим программным обеспечением, а так же выполняющих координацию вычислительных процессов, распределение аппаратных ресурсов между вычислительными процессами, управление доступом к процессам и ресурсам;

2) система управления базами данных (далее по тексту – СУБД) – совокупность программных и лингвистических средств (языков программирования и описания данных), обеспечивающих управление созданием и использованием баз данных (далее по тексту – БД);

3) сервер приложений – программная платформа, предназначенная для организации эффективного управления исполнением компонентов специального программного обеспечения. Сервер приложений также предназначается для эффективного и оперативного решения задач кластеризации, обеспечения отказоустойчивости и комплексирования сложных информационных систем.

Перечисленные выше компоненты общего назначения являются платформой для разработки и исполнения компонентов СПО.

Операционная система обеспечивает среду исполнения и реализации, как для прочих компонентов общего назначения, так и для специальных компонентов, предоставляя набор прикладных программных интерфейсов и решая задачи координации исполнения.

Система управления базами данных обеспечивает среду исполнения и реализации. СУБД предоставляет лингвистическое обеспечение для описания структуры хранилища и запросов к нему, обеспечивает транзакционность процессов изменения данных, а также непосредственно управляет процессам хранения и чтения данных на физическом уровне.

Сервер приложений обеспечивает среду исполнения и реализации для всех специальных компонентов. Сервер приложений предоставляет набор прикладных программных интерфейсов, посредством которых обеспечивается взаимодействие с ним сервисных компонентов СПО.

2.2.2 Состав и описание компонентов специального программного обеспечения

Специальное программное обеспечение включает в себя программный продукт COSOC и реализует основной функционал системы анализа:

— прием и верификацию входящих информационных сообщений о события, угрозах и инцидентах, соответствующих спецификации Common Alerting Protocol (CAP);

— запись сообщений в базу данных;

— поиск и выборку сообщений по заданным критериям;

СПО написано на языке Java. Комплекс разработан с использованием технологии J2EE, что обеспечивает ряд существенных преимуществ при создании сложных промышленных программных комплексов, а именно – высокая производительность, кроссплатформенность, надежность, масштабируемость, гибкость.

2.3 Взаимодействие между компонентами и внешними системами

Если вместе с COSOC поставляются дополнительные функциональные модули СПО (подсистема бизнес-аналитики, хранилище документов, документооборот, управление силами и средствами, и т.п.), то вместе с ними поставляется и интеграционная шина, и в этом случае модули СПО связаны между собой с через зарегистрированные в интеграционной шине web – сервисы.

Взаимодействие с внешними программами и АС осуществляется через REST API описание которого приведено в разделе 3.2.

3 Обязанности администратора и связанные с ними операции

В обязанноси администратора программного комплекса входит первоначальная устнановка, конфигурирование программых компонент и обеспечение работоспособности в процессе эксплуатации.

Создание моделей деятельности, настройка алгоритмов анализа и реагирования, заполнение классификаторов и т.д. не входит в обязанности адиминистратора, а выполняется консультантами-аналитиками либо разработчика решения, либо иной организации, занимающейся внедрением решения.

Программный комплекс основывается как на программных платформах с открытым кодом, так и на проприетарных решениях.

От администратора требуется знание соответствующих программных продуктов на уровне, позволяющем выполнить их установку и базовую настройку.

В данном разделе описаны основные задачи администратора по установке и обслуживанию ПК COSOC.

3.1 Развертывание программного комплекса COSOC

Для установки программного комплекса необхидим сервер с установленной операционной системой CentOS 6. Операционная система CentOS 6 включает в себя версию Python 2.4, но для установки COSOC требуется версия Python 2.7. Для того, чтобы поставить новую версию и оставить старую рекомендуем воспользоваться сторонним репозиторием IUS Community Project.

3.1.1 Установка Python 2.7

1) sudo rpm -ivh http://dl.iuscommunity.org/pub/ius/stable/Redhat/6/x86_64/epel-release-6-5.noarch.rpm

2) sudo rpm -ivh http://dl.iuscommunity.org/pub/ius/stable/Redhat/6/x86_64/ius-release-1.0-14.ius.el6.noarch.rpm

3) sudo yum clean all

4) sudo yum install python27

3.1.2 Установка COSOC

1) Направьте запрос по адресу <info@cos.ru> для получения ссылки на rpm дистрибутив программного комплекса. Скачайте и установие его при помощи утилиты yum.

2) Наберите в адресной строке браузера (или используйте curl в терминале): http://localhost:8080/CAPMessages/services/rest/test
В случае удачной установки и старта сервисов в ответ будет возвращено

```
{
"status": "Ok"
}
```

3.2 Использование REST API

3.2.1 Проверка работоспособности API

Для того чтобы убедиться, что сервис REST работает, наберите в адресной строке браузера (или используйте curl в терминале):
http://localhost:8080/CAPMessages/services/rest/test

В ответ должно быть возвращено

```
{
"status": "Ok"
}
```

3.2.2 Отправка CAP-сообщений

Чтобы отправить CAP сообщение (тестовое или отладочное) непосредственно с сервера, на котором установлен COSOC , используйте curl, например:
curl -v -X POST -d @cap.xml –header "Content-Type:text/xml;charset=UTF-8"
http://localhost:8080/CAPMessages/services/rest/insert

где "cap.xml" имя файла . Ответ будет возвращен в виде { "id": <id> }, где id = 0 в случае ошибки, и id > 0 в случае, если сообщение успешно сохранено.

Например:

```
{
  "id": 388920321792000
}
```

3.2.3 Выборка CAP-сообщений

Для выборки CAP сообщений используйте:
http://localhost:8080/CAPMessages/services/rest/find

В ответ будет возвращен массив значений некоторых полей сообщений (см. пример ниже) в формате json в порядке получения сервисом: последнее полученное сообщение имеет нулевой индекс.

Выборку сообщений можно фильтровать по следующим признакам:

— параметры времени (имеется в виду время когда сообщение было получено сервисом), формат - количество миллисекунд, истекших с 1 января 1970 года. timeStart - время первого сообщения (включительно) timeEnd - время последнего сообщения (исключительно, должен быть меньше timeStart).

— значения индексных полей indexes=index1,value1,index2,value2,..., indexN,valueN; если все значения index1..indexN разные, то это эквивалентно условию выборки index1=value1 AND index2=value2 AND ... indexN=valueN; если значения индексов совпадают, например index1=value1&index1=value2&index2=value3, то это эквивалентно условию выборки (index1=value1 OR index1=value2) AND index2=value3. В качестве индексов могут быть использованы следующие поля (*название индекса* - **поле в сообщении CAP**) :

source - **source**

audience - **audience**

onset - **onset**

ref_id - **references** (при условии что в CAP сообщении данное поле представлено числом)

inc_ref_id - **incidents** (при условии что в CAP сообщении данное поле представлено числом)

code - **code**

message_status - **status**

message_type - **msgType**

scope - **scope**

category - **category**

urgency - **urgency**

severity - **severity**

certainty - **certainty**

event - **event**

model - **model**

sender - **sender**

sender_name - **senderName**

Подробное описание значений полей и формата сообщения CAP приведены в Приложении №1 к данному руководству "Методические рекомендации по использованию Common Alerting Protocol для передачи информации о событиях и угрозах в программном комплексе ситуационного анализа COSOC".

— coordinates - массив координат области выборки, например: coordinates=58.50,31.10,58.55,31.10,58.55,31.18,58.50,31.18; область должна быть выпуклым многоугольником; для ускорения поиска рассматривается немного большая область, чем передаваемая в параметрах, и поэтому в выдачу могут попадать сообщения, находящиеся "недалеко"от выделенной области.

— индексы сообщений: indexStart - индекс первого сообщения (включительно), indexEnd - индекс последнего сообщения (исключительно, должен быть меньше indexStart); при смешивании параметров времени (timeStart, timeEnd) и индексов (indexStart, indexEnd) в одном запросе в

выборку попадут CAP сообщения, попадающие в пересечение этих двух множеств.

— maxCount - макс кол-во сообщений в выдаче (не более 65535)

Примеры:
http://localhost:8080/CAPMessages/services/rest/find?coordinates=58.50,31.10,58.55,31.10,58.55,31.18,58.50,31.18&indexes=sender,639274e1-daad-4f67-a68a-5c2bf45123df&indexStart=387553692928000&indexEnd=387402165248000&maxCount=1

http://localhost:8080/CAPMessages/services/rest/find?&indexes=scope,Public,category,Meteo

При вызове без параметров сервис возвращает последние 1000 сообщений.

Например:
http://localhost:8080/CAPMessages/services/rest/find/


```
[{
  "id":394699598080000,
  "scope":"Public",
  "code":"Publish",
  "sentDate":"2017-12-26T09:17:08+00:00",
  "infos":[
  {
    "category":"Met",
    "urgency":"Immediate",
    "severity":"Moderate",
    "certainty":"Observed",
    "headline":"Request #1",
    "event":"",
    "model":"",
    "coordinates":[
        [56.001174,92.818531,0.1]
```

```
    ],
    "eventCodeMap":{},
    "parametersMap":
    {
      "Status":"Actual",
      "Type":"Problem",
      "E-mail":"mail@example.com",
      "Priority":"Middle",
      "Phone":"+71234567890"
    },
    "senderName":"Ivanov I.I.",
    "description":"Meteo request",
    "onset":"",
    "audience":"all",
    "effective":"2017-12-26T05:22:08+00:00",
    "resources": null,
    "state":1,
    "state1":0,
    "voteCount":0,
    "replyCount": 0,
    "source": "EIAS",
    "incidentRefId":0,
    "refMessageId":0,
    "messageId":394699598080000,
    "senderId":"EAIS",
    "messageStatus":"Actual",
    "messageType":"Alert"
}]
```

3.2.4 Получение оригинала CAP-сообщения

Для получения "почти"оригинала CAP'а в формате xml используйте:

http://localhost:8080/CAPMessages/services/rest/<id>/xml

Например:

http://localhost:8080/CAPMessages/services/rest/388727053568000/xml

От оригинального сообщения отличается только заполненным полем identifier:

```
<alert xmlns="urn:oasis:names:tc:emergency:cap:1.2">
    <identifier>394699598080000</identifier>
    <sender>EIAS</sender>
    <sent>2017-12-26T09:17:08+00:00</sent>
    <status>Actual</status>
    <msgType>Alert</msgType>
    <source>EIAS</source>
    <scope>Public</scope>
    <code>Publish</code>
    <info>
      <parameter>
      <valueName>LIKE</valueName>
      <value>0</value>
      </parameter>
    <parameter>
      <valueName>RecourseID</valueName>
      <value>123</value>
    </parameter>
    <parameter>
      <valueName>Type</valueName>
      <value>Problem</value>
    </parameter>
    <parameter>
      <valueName>Priority</valueName>
      <value>Normal</value>
    </parameter>
    <parameter>
      <valueName>Status</valueName>
      <value>Actual</value>
    </parameter>
    <language>en_EN</language>
    <category>Met</category>
    <responseType>None</responseType>
    <urgency>Immediate</urgency>
    <severity>Moderate</severity>
    <certainty>Observed</certainty>
    <audience>all</audience>
    <effective>2017-12-26T05:22:08+00:00</effective>
    <senderName>Ivanov I.I.</senderName>
    <headline>Request #1</headline>
```

```
    <description>Meteo request</description>
    <parameter>
      <valueName>Phone</valueName>
      <value>+71234567890</value>
    </parameter>
    <parameter>
      <valueName>E-mail</valueName>
      <value>mail@example.com</value>
    </parameter>
    <area>
      <areaDesc>Request point</areaDesc>
      <circle>56.001174,92.818531 0.0</circle>
    </area>
  </info>
</alert>
```

3.2.5 Получение сообщения в формате json

В формате json возвращаются поля, обязательные с точки зрения спецификации CAP, и некоторое количество опциональных и служебных полей, используемых для хранения дополнительных данных, ассоциированных с CAP сообщением (см. пример ниже).

Для получения полей CAP-сообщения в формате json используйте:

http://localhost:8080/CAPMessages/services/rest/<id>/json

Например:

http://localhost:8080/CAPMessages/services/rest/388727053568000/json

```
{
  "id":394699598080000,
  "scope":"Public",
  "code":"Publish",
  "sentDate":"2017-12-26T09:17:08+00:00",
  "infos":[
  {
    "category":"Met",
    "urgency":"Immediate",
    "severity":"Moderate",
```

```
    "certainty":"Observed",
    "headline":"Request #1",
    "event":"",
    "model":"",
    "coordinates":[
        [56.001174,92.818531,0.1]
    ],
    "eventCodeMap":{},
    "parametersMap":
    {
      "Status":"Actual",
      "Type":"Problem",
      "E-mail":"mail@example.com",
      "Priority":"Middle",
      "Phone":"+71234567890"
    },
    "senderName":"Ivanov I.I.",
    "description":"Meteo request",
    "onset":"",
    "audience":"all",
    "effective":"2017-12-26T05:22:08+00:00",
    "resources":
    [
        {
            "resourceDesc":"",
            "mimeType":"",
            "size":null,
            "uri":null,
            "derefUri":"",
            "digest":null
        }
    ]
  ],
```

```
  "state":1,
  "state1":0,
  "voteCount":0,
  "replyCount": 0,
  "source": "EIAS",
  "incidentRefId":0,
  "refMessageId":0,
  "messageId":394699598080000,
  "senderId":"EAIS",
  "messageStatus":"Actual",
  "messageType":"Alert"
}
```

Поля state, state1, voteCount, replyCount не являются частью CAP спецификации и поэтому должны игнорироваться.

4 Сообщения об ошибках

При возникновении сообщения о системной ошибке в консоли оператора подсистемы программного комплекса следует следовать рекомендациям соответствующих руководств общего программного обеспечения.

При возникновении сообщения об ошибке, исходящей от СПО, связаться со службой технической поддержки разработчика АО "ЦОСиВТ" по телефону 8-800-333-0092 или по e-mail: <info@cos.ru>, указав текст сообщения, приложив копию экрана, сообщив логин пользователя, скопировав адрес страницы, на которой произошла ошибка.