

# Мета-данные в REST модели

Дмитрий Намиот

Лаборатория Открытых  
Информационных  
Технологий  
[dnamiot@gmail.com](mailto:dnamiot@gmail.com)

Москва 2015

# Содержание

- REST модель
- REST vs. SOA
- Метаданные и их назначение
- Формы представления метаданных в REST
- Дискуссия

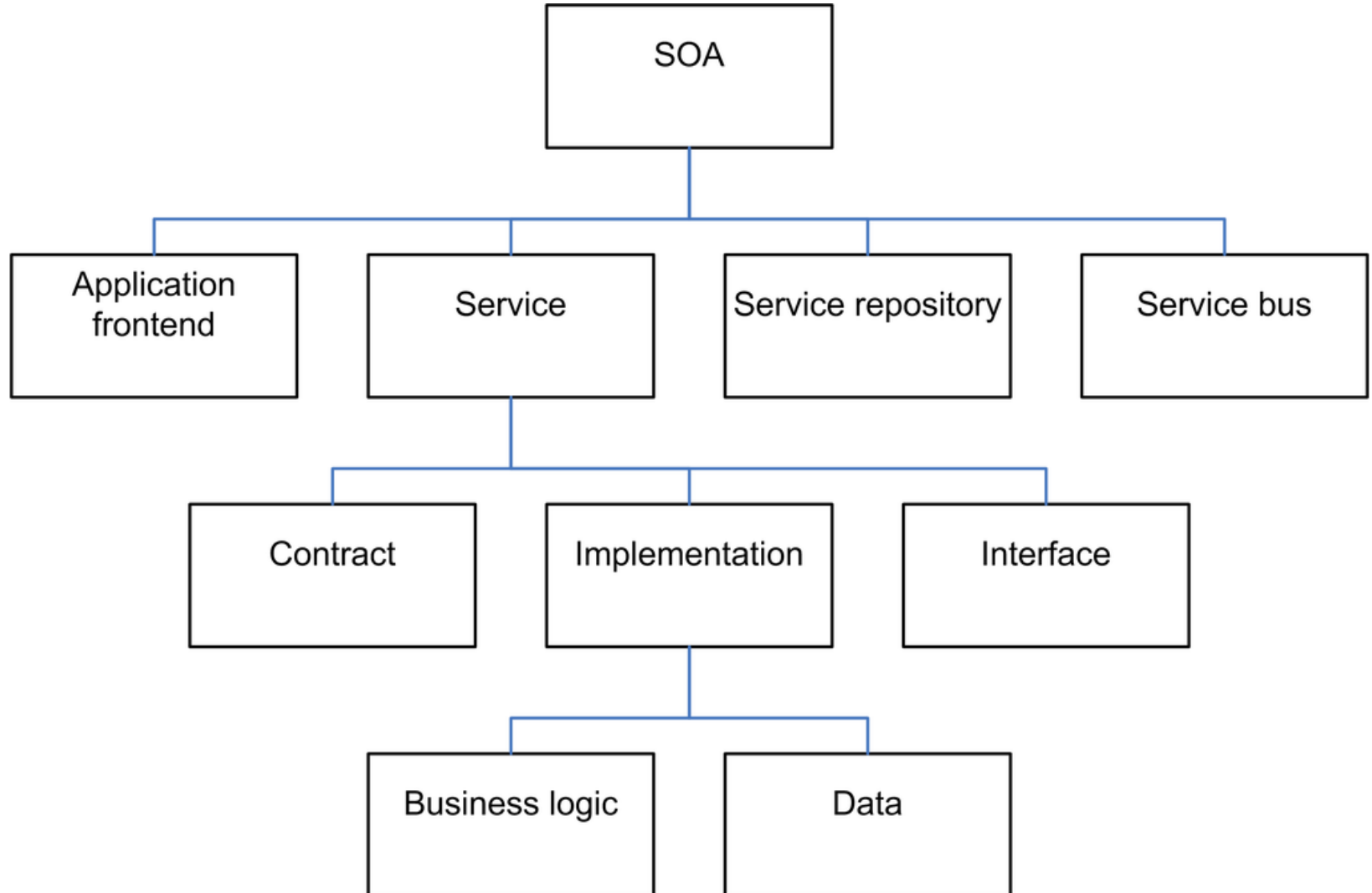
# REST

- Малый набор базовых принципов
- Унифицированный интерфейс
- Отсутствие необходимости поддержки состояний
- Простая структура запросов (self-descriptive messages).

# REST

- Клиент-серверная архитектура
- Сервер не обязан сохранять информацию о состоянии клиента
- Запрос клиента явно содержит указание о возможности кэширования ответа и получения ответа из кэша
- Возможность взаимодействия через промежуточные узлы
- Унифицированный программный интерфейс сервера (URI как пример).

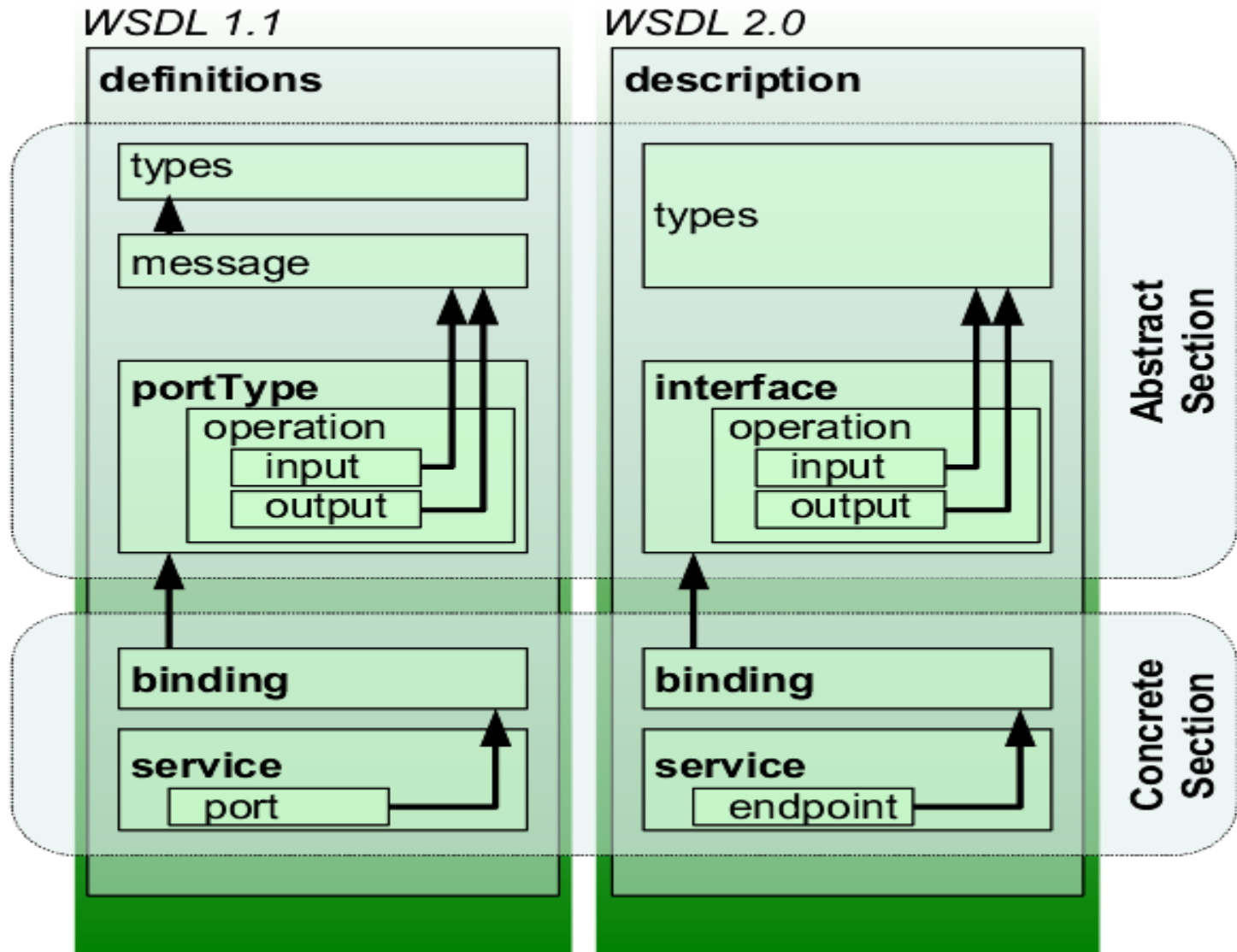
# SOA



# Метаданные

- Данные о данных
- Метаданные в REST:
- отсутствуют в силу унификации интерфейса
- Метаданные в SOA:
- необходимость описывать различные ресурсы
- поддержка персонализации доступа (обращений)

# WSDL



# Использование метаданных

- Задачи:
- Автоматизация программирования (генерации программ, тестирования, верификации)
- Документирование
  
- Области применения
- Много разных интерфейсов: M2M, IoT
- Непрерывность обслуживания (SLA): телеком



# Метаданные в REST

- Описание точек доступа
- URLs для запросов
- Пути для ресурсов
- Методы для доступа к ресурсам (HTTP команды)
- Параметры в запросах.
- Заголовки HTTP
- Коды возвратов и сообщения об ошибках.
- Версии API
- Форматы для результатов запросов.  
(JSON, JSONP, XML)

# Псевдо-Метаданные в REST

- URL запроса используется для кодирования
- Twitter API:

[https://api.twitter.com/1.1/statuses/mentions\\_timeline.json](https://api.twitter.com/1.1/statuses/mentions_timeline.json)

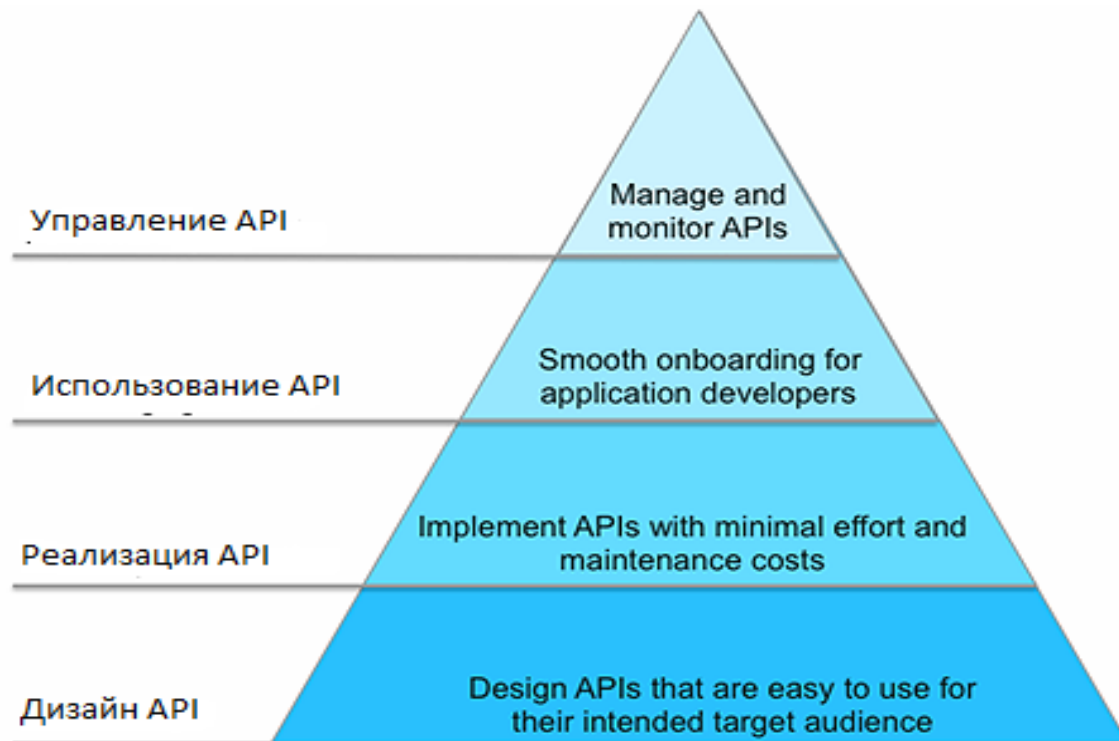
- *1.1* – это и есть номер версии.
- *.json* – формат отклика
- Нет возможности узнать про сам URL – только из документации

# Поддержка Метаданных в REST

- Разметка кода:
- Swagger, API Blueprint
- Основной результат:
- готовая документация
- статический генератор
- Аналогии:
- JavaDoc

# Поддержка Метаданных в REST

- Платформы для API:
- Intel Mashery, Anypoint Platform



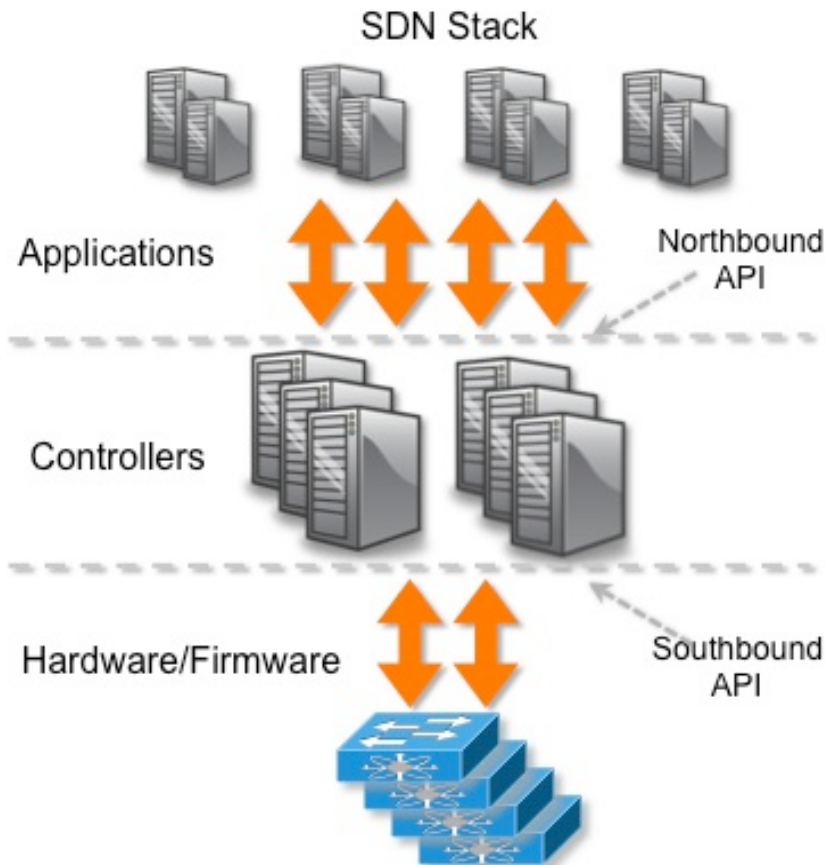
# Поддержка Метаданных в REST

- Языки описания
- HAL, RSDL, WADL
- Web Application Description Language (WADL):
- XML для описания веб-приложений
- Наиболее близкий аналог WSDL
- WADL предложен как стандарт W3C
- Есть средства генерации кода по описанию и описания по коду (Oracle, Java, JSR-311)

# Дискуссия

- Стандарта де-факто: нет
- Сообществом разработчиков WADL не принят
- Java среда наиболее близка к “мета-данным в REST”
- Задачи автоматизации программирования останутся актуальны
- M2M (IoT) приложения не достигли критических масштабов по разнообразию устройств ?
- Wearables, BYOD могут изменить картину
- Возможность вертикальных решений

# SDN



- GET /v2.0/networks?limit=2
- Accept: application/json
- Отклик:
- {
- "networks":[
- {
- "status":"ACTIVE",
- "subnets":[ ...