Эволюция идеологии систем удалённого управления на примере IPMI и RedFish

IPMI

Intelligent
Platform
Management
Interface

http://www.intel.com/design/servers/ipmi/

1998 год — представлена версия 1.0

2004 год — представлена версия 2.0, являющаяся
(с незначительными дополнениями)
основной по сей день

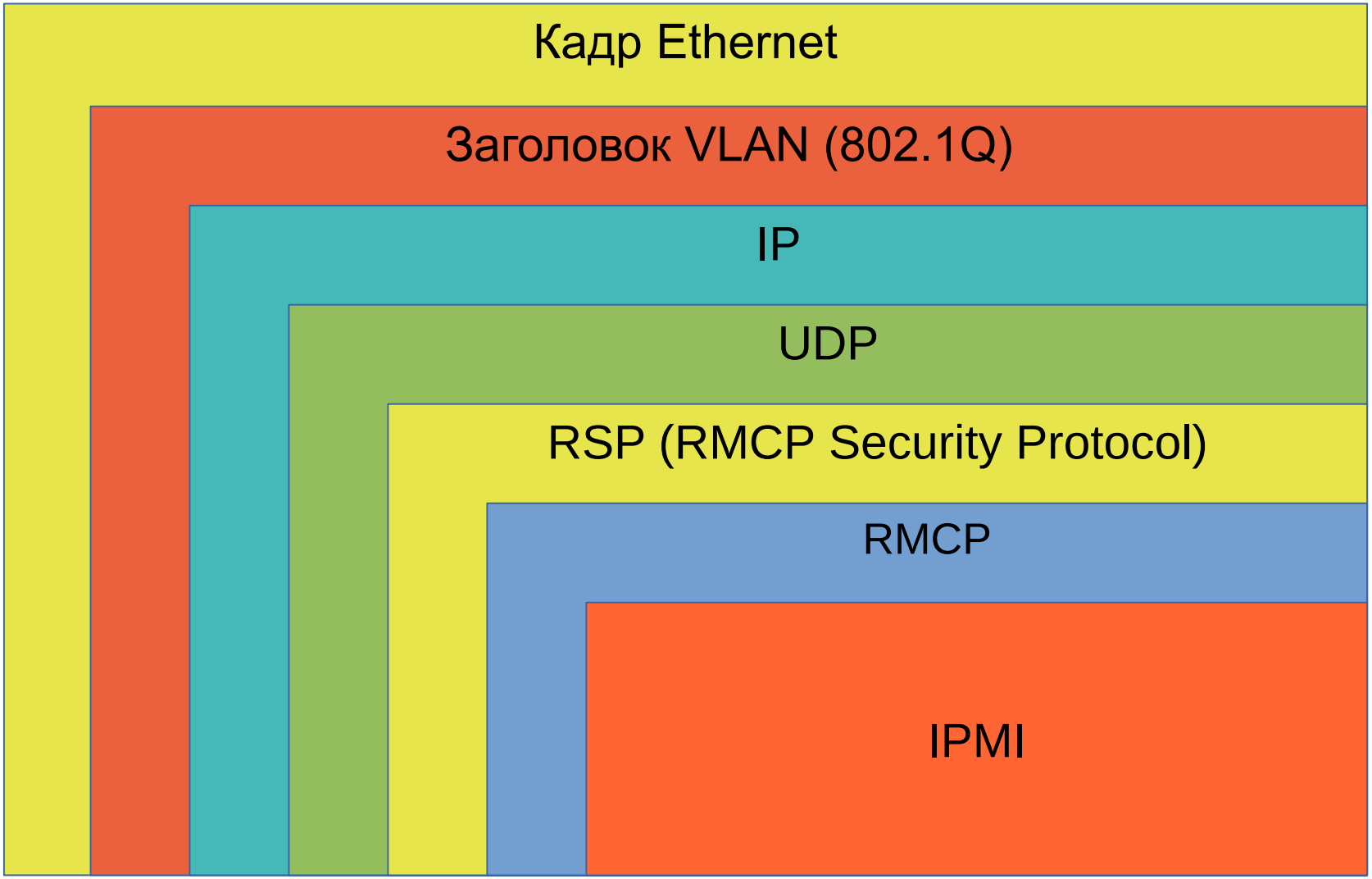
IPMI
версия 1.0 — 1998 год

Компьютер 1997 года

Процессор Slot 1

Частоты ЦП — порядка 200-300 МГц
Объём памяти — порядка 64 Мб

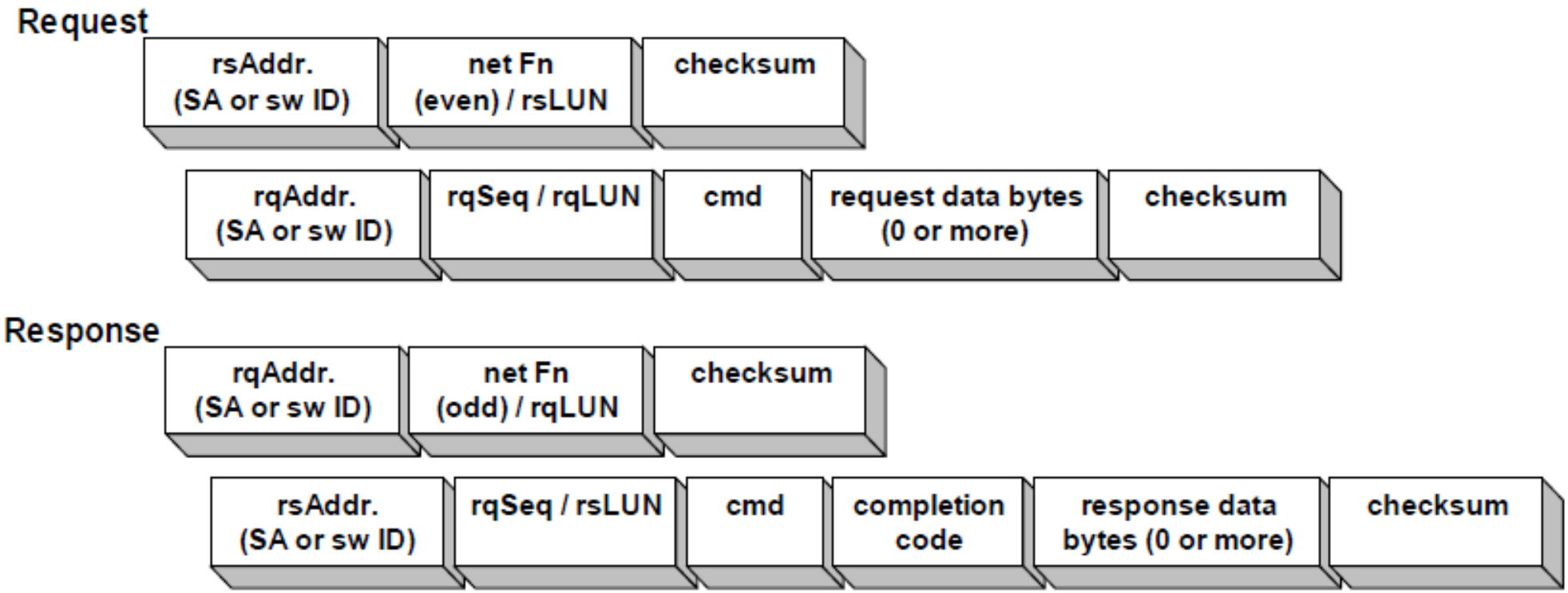
Инкапсуляция пакета IPMI



Фрагмент формата пакета RMCP с IPMI содержимым

RSP Header	Session ID		4		
	Session Sequence #		4		
RMCP Header	Version	1	1	1	06h (ASF 2.0)
	Reserved	1	1	1	
	RMCP Seq #	1	1	1	FFh for IPMI[2]
	Class of Message	1	1	1	07h for IPMI
IPMI Session Header	Auth Type / Format	1		1	[7:4] - reserved [3:0] - Authentication Type / Format 0h = none 1h = MD2 2h = MD5 3h = reserved 4h = straight password / key 5h = OEM proprietary 6h = Format = RMCP+ (IPMI v2.0 only) all other = reserved

Формат запроса и ответа IPMI



RedFish

http://www.dmtf.org/standards/redfish

2015 год — представлена первая версия

- Версии API обновляются постоянно,
- на текущий момент доступна версия 1.0.4
- Версии схемы данных обновляются раз в полгода,
- в данный момент выпущена третья версия (2016.2)
- API и схемы данных обладают обратной совместимостью

RedFish - разработчики

Leadership Companies

Supporting Companies

AMI, Cisco, Fujitsu, Western Digital, Huawei, IBM, Insyde Software, Mellanox, NetApp, Oracle, Microsemi, Qualcomm, Seagate

Industry Alliance Partners

- OpenCompute Project
- UEFI - Collaborating on Firmware Update and Host Interface work
- SNIA – Collaborating on Storage modeling/alignment between SSM and Redfish
- TGG – Pursuing relationship to work on Power/Cooling (existing DMTF Alliance Partner)

Зачем понадобился RedFish?

Каким должен быть современный интерфейс удалённого управления?

- Простой в освоении и использовании
- Максимально открытый
- Безопасный
- Способный управлять расширяемыми системами

Индустрия не хочет обучать персонал новому набору инструментов для выполнения стандартных задач управления

RedFish — решения и результат

- REST-интерфейс
- Широкое использование существующих стандартов,
- широко применяемых в индустрии: HTTPS, JSON
- Возможность использования в качестве клиентов
- браузеров и GUI-ориентированных приложений
- Обмен данными в человеко-читаемом формате
- облегчает разработку и отладку

Client Python code

```
rawData = urllib.urlopen('https://<ip_addr>/redfish/v1/Systems/1')
jsonData = json.loads(rawData)
print( jsonData['SerialNumber'] )
```

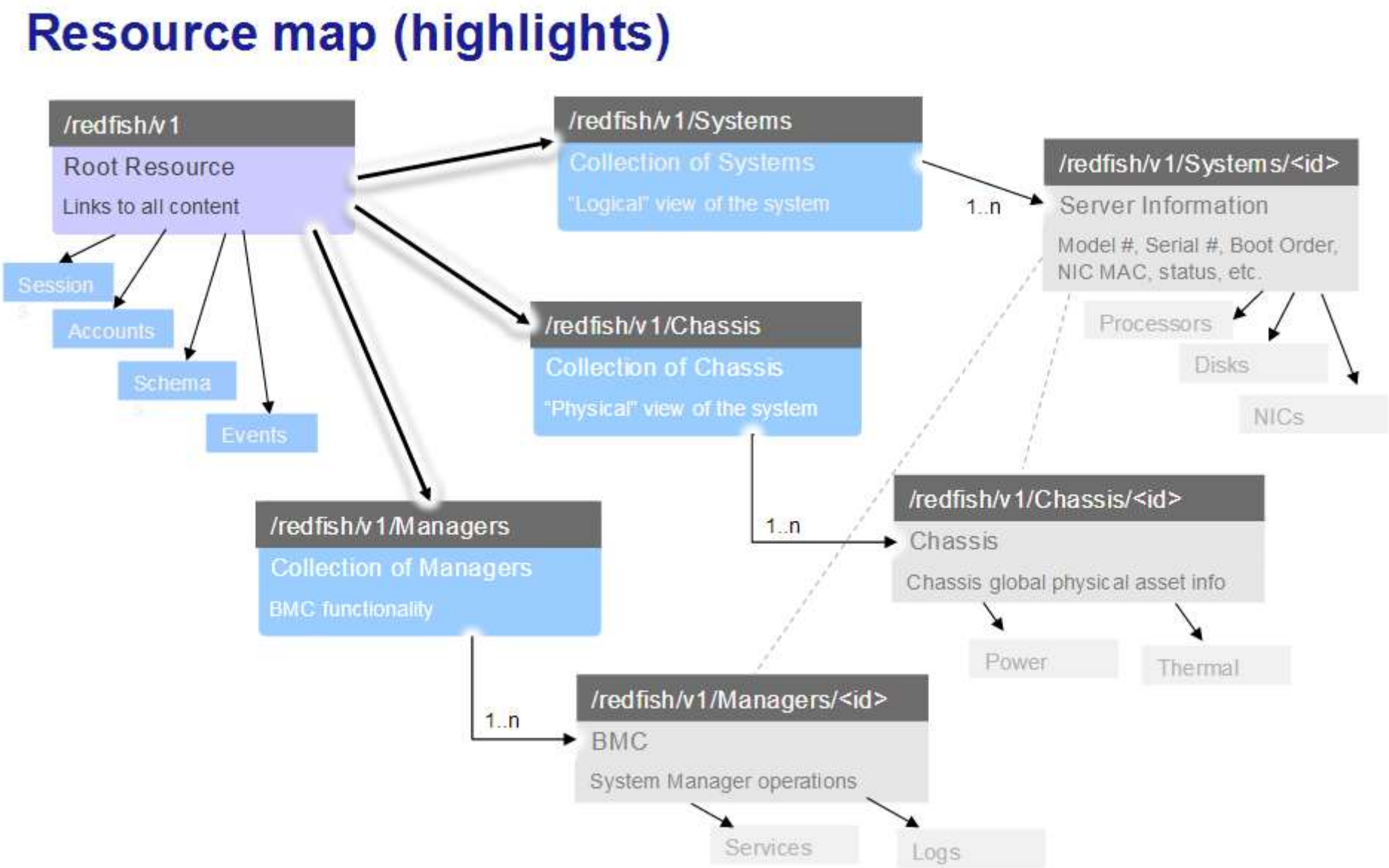
Output

```
1A87CA442K
```

RedFish
причины выбора компонентов

- REST
- Стремительно набирает популярность, быстро вытесняя
- SOAP как протокол и RPC как идеологию
- HTTPS
- Хорошо знаком администратором, безопасен,
- издержки внедрения минимальны (конфигурация
- оборудования)
- JSON
- «The fat-free alternative to XML»
- Компактен, универсален, хорошо читается человеком

RedFish — схема данных



www.dmtf.org

IPMI и RedFish — 17 лет эволюции

IPMI

- Монолитный стандарт, включающий в себя протокол обмена, схему данных и спецификацию шины
- Низкоуровневый протокол, определяет формат общения вплоть до отдельных битов
- Допускает прямое взаимодействие клиента с данными, использует контекст при взаимодействии сервера с клиентом

IPMI и RedFish — 17 лет эволюции

RedFish

- Составной стандарт, использующий HTTP для передачи данных, XML для описания их структуры
- Высокоуровневый стандарт, определяет формат общения и структуру данных не напрямую, а используя метаязык использованных стандартов
- Клиент не имеет прямого доступа к данным, между обменами не сохраняется контекст