Ch4. 数据编码与演化

Zhiyi Xiao

背景

- 数据格式或模式变化
- 调整应用程序代码
- 代码更迭
- 服务器端：执行滚动升级（分阶段发布）
- 客户端：不会马上安装更新
- 新旧版本代码在系统内共存

双向兼容

- 向后兼容：新代码可以读取旧数据
- 向前兼容：旧代码可以读取新数据

数据编码格式

- 内存
- 写入文件/通过网络发送
- 序列化/编码
- 反序列化/介些

语言特定的形式

- 编程语言内置函数
  - Java：java.io.Serializable
  - Python：pickle
- 存在问题

JSON、XML

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<俄语 լեզու="ռուսերեն">данные</俄语>
```

Binary Encoding

```
{

  "userName": "Martin",
  "favoriteNumber": 1337,
  "interests": ["daydreaming", "hacking"]

}
```

MessagePack

Byte sequence (66 bytes):

```
83 | a8 | 75 73 65 72 4e 61 6d 65 | a6 | 4d 61 72 74 69 6e | ae | 66 61
76 6f 72 69 74 65 4e 75 6d 62 65 72 | cd | 05 39 | a9 | 69 6e 74 65
72 65 73 74 73 | 92 | ab | 64 61 79 64 72 65 61 6d 69 6e 67 | a7 | 68
61 63 6b 69 6e 67
```

Breakdown:

object (3 entries)	string (length 8)	u s e r N a m e	string (length 6)	M a r t i n
83	a8	75 73 65 72 4e 61 6d 65	a6	4d 61 72 74 69 6e
	string (length 14)	f a v o r i t e N u m b e r		
	ae	66 61 76 6f 72 69 74 65 4e 75 6d 62 65 72		
	uint16	1337	string (length 9)	i n t e r e s t s
	cd	05 39	a9	69 6e 74 65 72 65 73 74 73
array (2 entries)	string (length 11)	d a y d r e a m i n g		
92	ab	64 61 79 64 72 65 61 6d 69 6e 67		
	string (length 7)	h a c k i n g		
	a7	68 61 63 6b 69 6e 67		

Thrift and Protocol Buffers

```
struct Person {
1: required string          userName,
2: optional i64             favoriteNumber,
3: optional list<string>    interests
}
```

```
message Person {
required string user_name = 1;
optional int64 favorite_number = 2;
repeated string interests = 3;
 }
```

Thrift BinaryProtocol

Byte sequence (59 bytes):

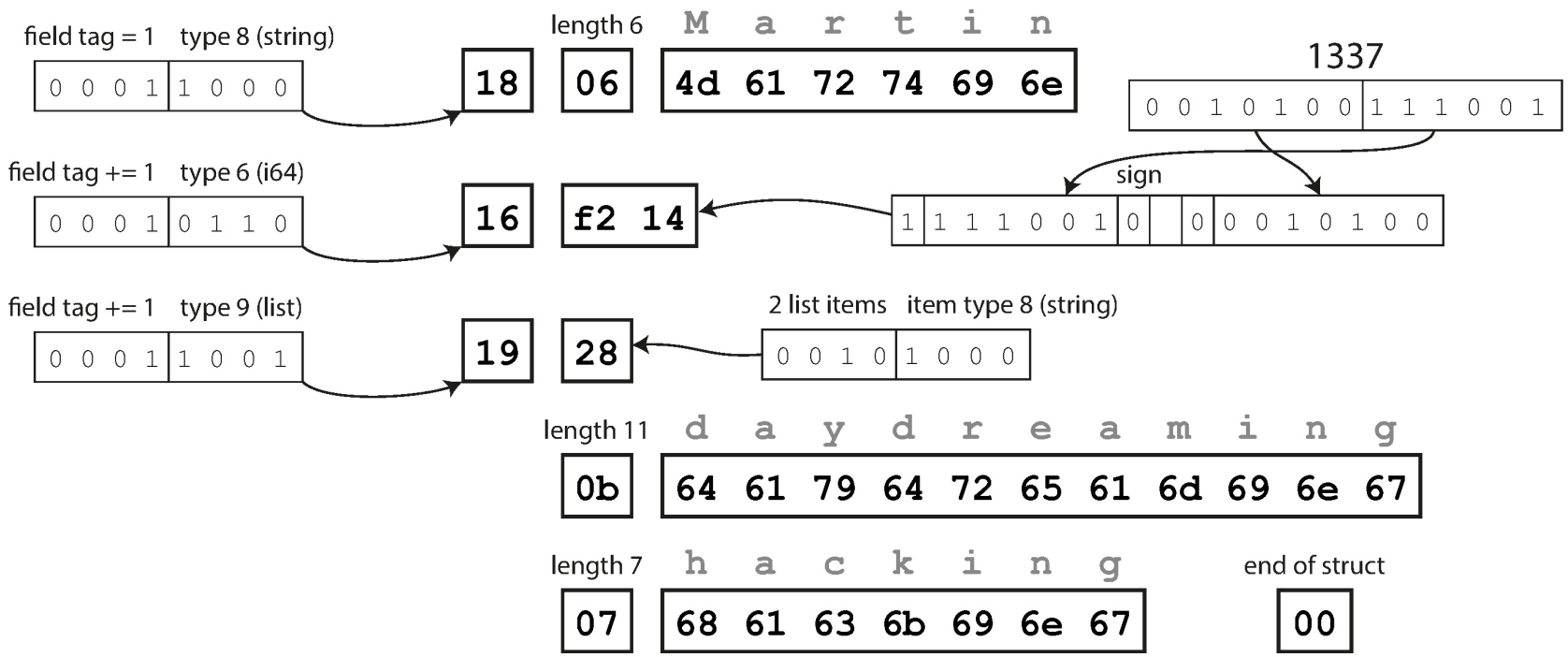
| | | |

Thrift CompactProtocol

Byte sequence (34 bytes):

18	06	4d	61	72	74	69	6e	16	f2	14	19	28	0b	64	61	79	64	72	65	61	6d	69	6e	67	07	68	61	63	6b	69	6e	67	00

Breakdown:

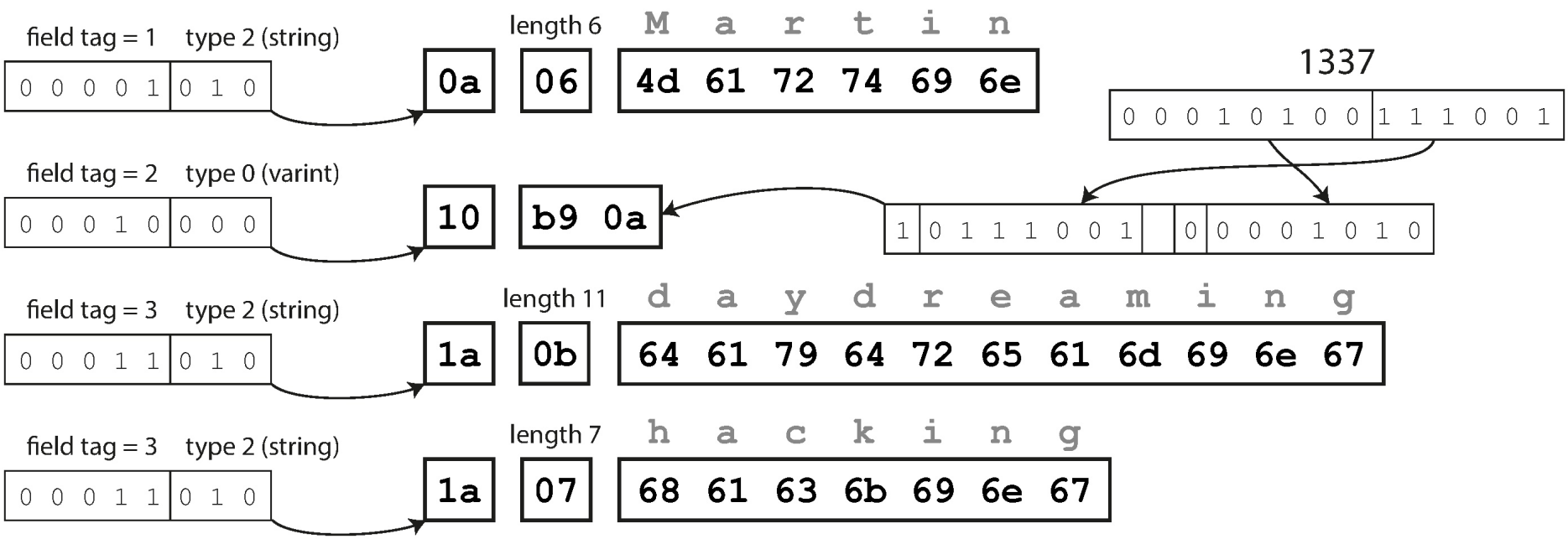


Protocol Buffers

Byte sequence (33 bytes):

0a	06	4d	61	72	74	69	6e	10	b9	0a	1a	0b	64	61	79	64	72	65	61	6d	69	6e	67	1a	07	68	61	63	6b	69	6e	67

Breakdown:



Avro

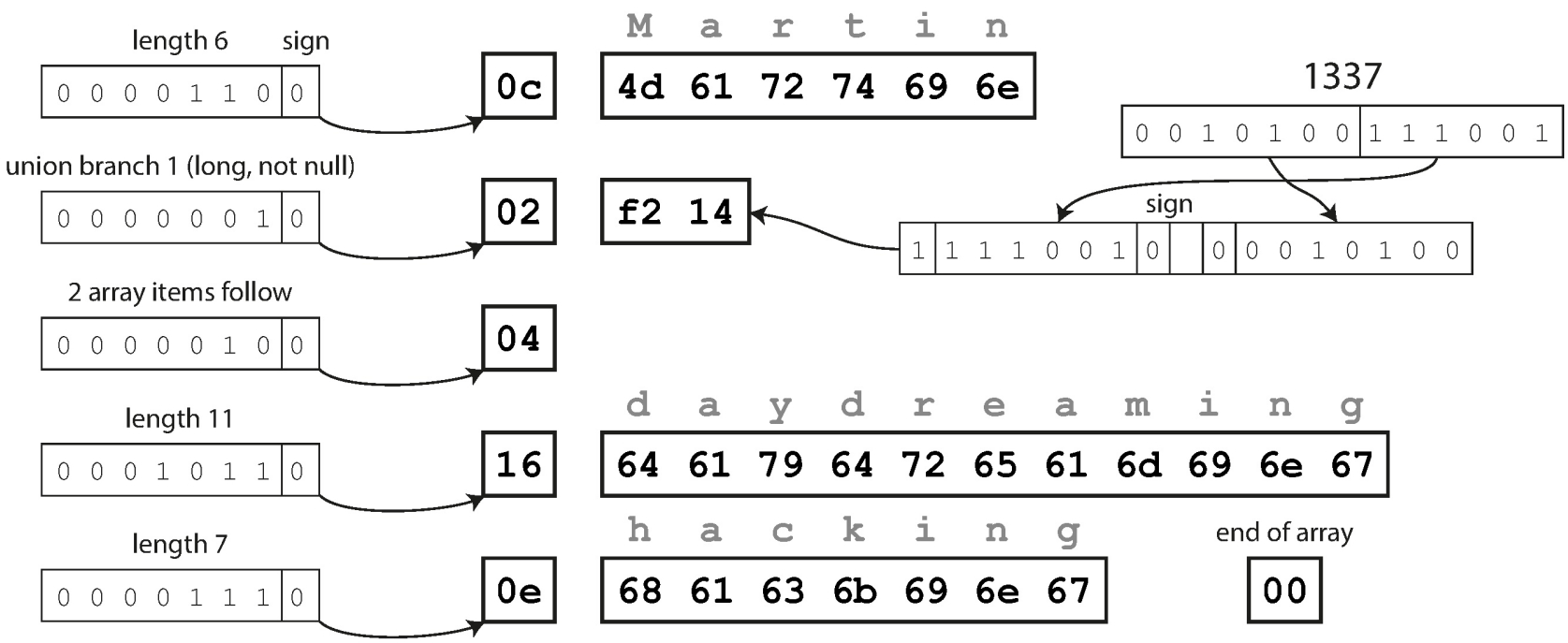
```
record Person {
string                          userName;
union { null, long }            favoriteNumber = null;
array<string>                   interests;
}
```

Avro

Byte sequence (32 bytes):

0c	4d	61	72	74	69	6e	02	f2	14	04	16	64	61	79	64	72	65	61	6d	69	6e	67	0e	68	61	63	6b	69	6e	67	00

Breakdown:



Writer's schema for Person record

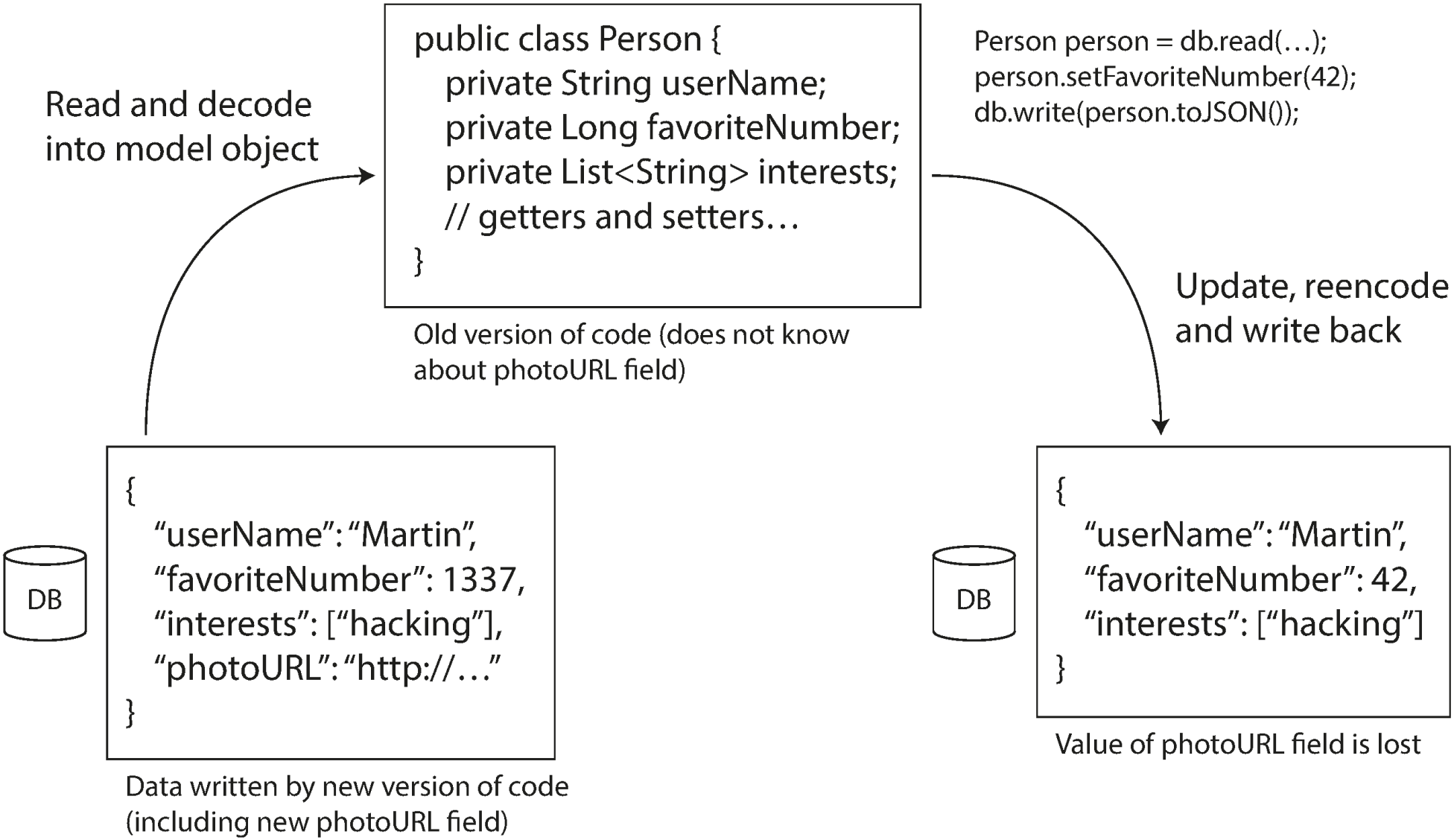
Datatype	Field name
string	userName
union {null, long}	favoriteNumber
array<string>	interests
string	photoURL

Reader's schema for Person record

Datatype	Field name
long	userID
union {null, int}	favoriteNumber
string	userName
array<string>	interests

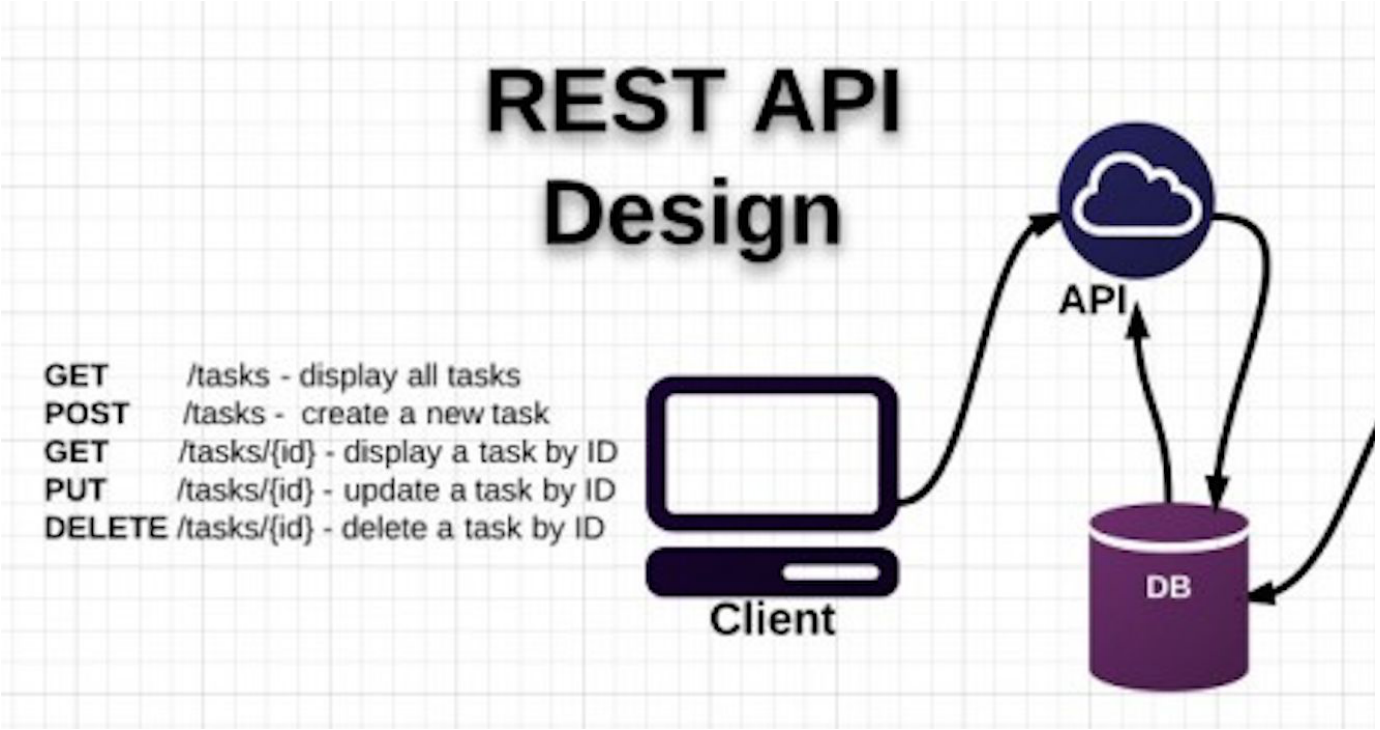
数据流模式

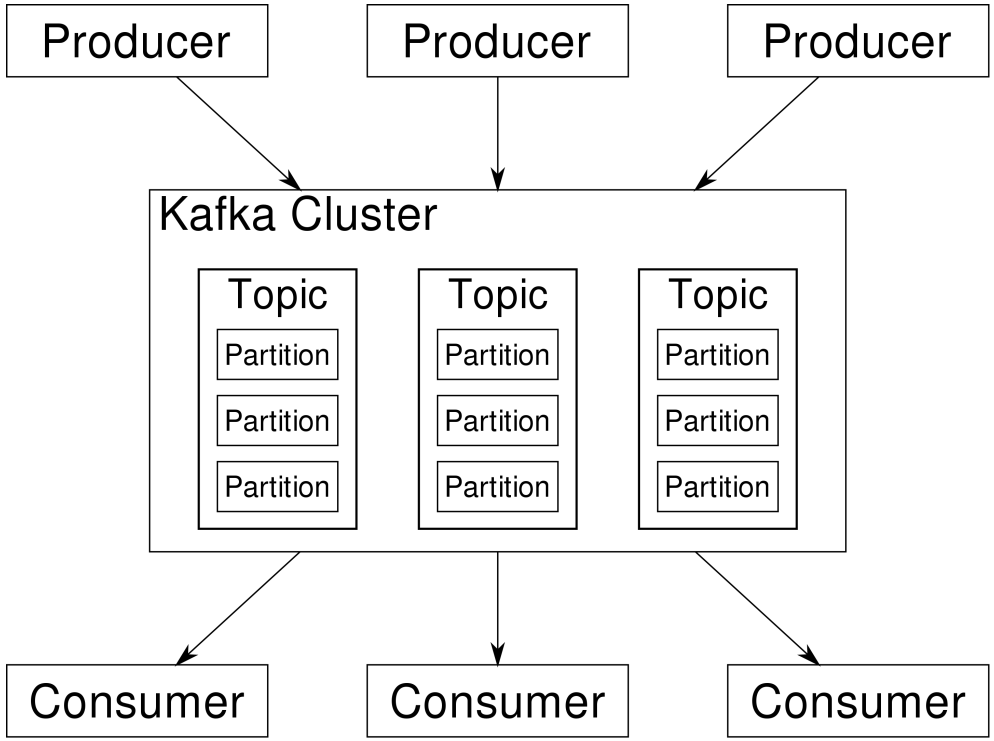
- 数据库
- 服务调用
- 异步消息传递


Read and decode
into model object
public class Person {
private String userName;
private Long favoriteNumber;
private List<String> interests;
// getters and setters…
}
Old version of code (does not know
about photoURL field)
Person person = db.read(…);
person.setFavoriteNumber(42);
db.write(person.toJSON());
Update, reencode
and write back
DB
{
"userName": "Martin",
"favoriteNumber": 1337,
"interests": ["hacking"],
"photoURL": "http://…"
}
Data written by new version of code
(including new photoURL field)
DB
{
"userName": "Martin",
"favoriteNumber": 42,
"interests": ["hacking"]
}
Value of photoURL field is lost

基于服务的数据流服务

- 网络服务
  - REST
  - SOAP
- RPC



基于消息传递的数据流

- 优势
- Apache Kafka