# Облачные вычислительные среды

Писковский Виктор Олегович

# План

Введение

1. Распределенные реестры и криптометоды (пороговая подпись, гомоморфное шифрование, доказательство с «нулевым» знанием)
2. Оценка защищенности архитектур распределенных информационных систем
3. Алгоритмы консенсуса
4. Использование цифровых водяных знаков и небайесовские задачи распознавания
5. Сериализация и проактивная конфигурация сетей

# Облачные вычисления. Определение

Облачные вычисления (Cloud computing) — модель выполнения вычислений удаленно, с помощью сетевого доступа «по запросу» к конфигурируемым вычислительным ресурсам с целью получения вычислительных услуг, которые предоставляются по возможности быстро и практически автоматически, то есть с минимальными усилиями со стороны администраторов этих ресурсов.

*«The NIST Definition of Cloud Computing»: Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models*

# Характеристики ОВС

1. Возможность самообслуживания без участия человека со стороны провайдера (*On-demand self-service*)
2. Наличие широкополосного доступа к сети (*Broad network access*)
3. Сосредоточенность ресурсов на отдельных площадках для их эффективного распределения (*Resource pooling*)
4. Быстрая масштабируемость — ресурсы могут неограниченно выделяться и высвобождаться с большой скоростью в зависимости от потребностей (*Rapid elasticity)*
5. Управляемый сервис — система управления облаком автоматически контролирует и оптимизирует выделение ресурсов, основываясь на измеряемых параметрах сервиса (размер системы хранения, ширина полосы пропускания, число активных пользователей и т. д.) (*Measured service*)

https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%9E%D0%B1%D0%BB%D0%B0%D1%87%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_(Cloud_computing)

# Услуги ОВС

1. SaaS
2. PaaS
3. IaaS

- *MSaaS (Managed Sw aaS)*
- *SECaaS (Security aaS)*
- *VPNaaS (VPN aaS)*
- *DTaaS (Desktop aaS)*
- *DBaaS (Data Base aaS)*
- *STaaS (Storage aaS)*
- *LaaS (Logging aaS)*
- *XaaS (Everytnig aaS)*
- *HaaS (Hw aaS)*
- *DaaS (Data aaS)*
- *DCaaS (Data Center aaS)*
- *MBaaS (Mobile Backend aaS)*
- *IT(M)aaS (IT Management aaS)*
- *FaaS (Function/Serverless) aaS)*
- *BPMaaS (Business Process Management aaS)*
- *RaaS (Ransomware aaS)*
- *…*

# Типы ОВС

1. Частное облако (*Private cloud*)
2. Общественное облако (*Community cloud*)
3. Публичное (*Public cloud*)
4. Гибридное (*Hybrid cloud*)

# Архитектуры безопасности в системах цифровой экономики

Распределенные реестры и криптометоды (пороговая подпись, гомоморфное шифрование, доказательство с «нулевым» знанием)

# Регистрация фактов доступа к защищаемым данным

Требования к средству регистрации фактов доступа:

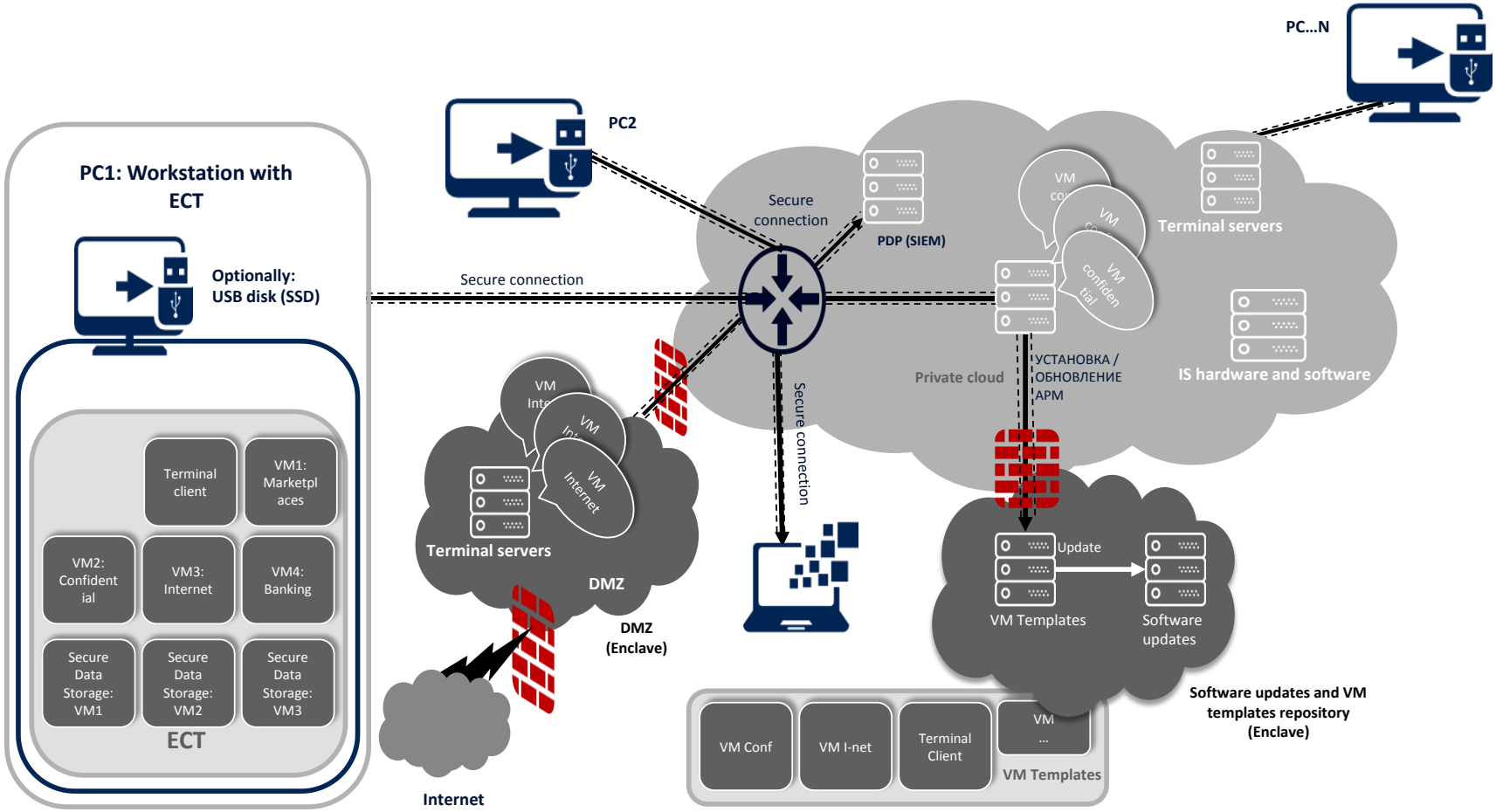
- децентрализация доступа, хранения и учета, отсутствие логического центра управления
- исключение фактов фальсификации регистрируемой информации
- возможность деанонимизации хранимой журнальной информации должна быть только у собрания представителей коллегиального органа, обладающего соответствующими полномочиями, минимальное количество таких представителей регулируется соглашением,
- возможность для владельцев информации получить данные о наличии фактов и времени доступа к охраняемым данным.

# Распределенные реестры и привлекаемые технологии

| Технология | Назначение |
|---|---|
| Доказательство с нулевым разглашением | Возможность подтвердить факт доступа и не допустить возможности достоверно подтвердить кто и как |
| Конфиденциальное вычисление | Интерактивное вычисления без предоставления к данным источников |
| Гомоморфное шифрование | Облачные вычисления без знания содержимого |
| Распределённая схема подписи | Независимость участников при постановке подписи, не интерактивный, асинхронный протокол выдачи проекций секретного ключа, не требует участия дилера. Для проверки достаточно наличия, порогового количества подписантов |

# Оценка защищенности архитектур распределенных информационных систем

# Comprehensive information security for private cloud computing environments



# Алгоритмы консенсуса

Типы консенсуса и модели DLT

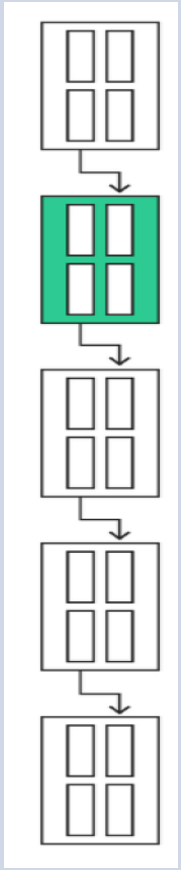
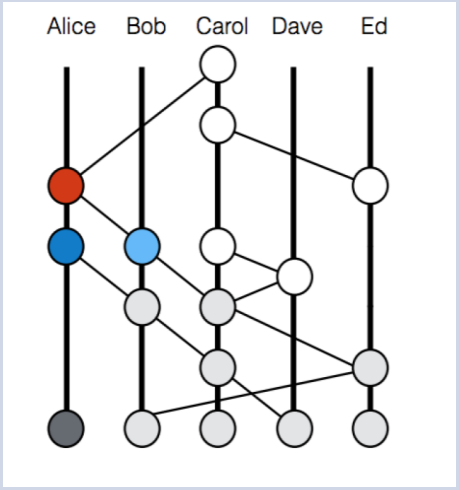
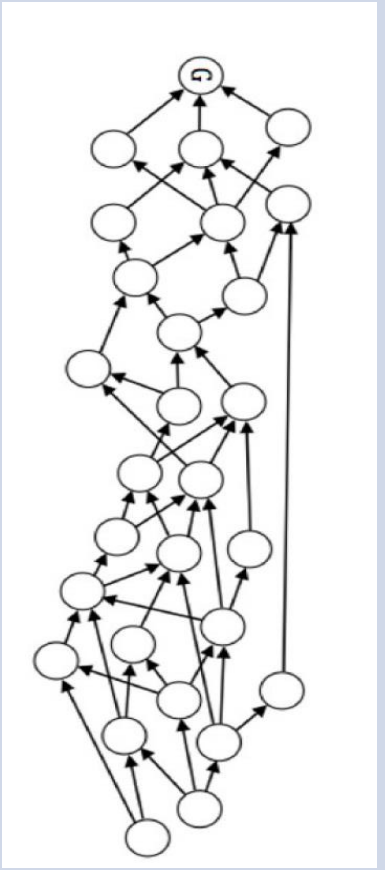
# Типы консенсуса и модели DLT

| № п.п | Модель DLT | Протокол достижения консенсуса | Производительность (TPS) | Примечание |
|---|---|---|---|---|
| 1 | BlockChain | Proof of Work (PoW) | 10 | Регулируется соглашением |
| 2 | | Proof of Storage (PoSt) | 100-200 | |
| 3 | | Proof of Stake (PoS) | 100-200 | |
| 4 | | Byzantine Fault Tolerance (BFT) | 1 тыс | Отдельные решения до 1 млн |
| 5 | HashGraph | Asynchronous BFT (ABFT) | 250 - 350 тыс. | SWIFT-VISA (50 тыс. TPS) |
| 6 | DAG | BlockDAG/TxDAG | > 1 млн | |

# Модель DLT

| Технология | Модель DLT | Комментарий |
|---|---|---|
| Связный список | Blockchain | Связный список вытянутых в цепочку блоков, связанных соотношением один к одному |
| Направленный ациклический граф | HashGraph | Каждый узел хранит свою историю "событий“. Протокол "слухов" (gossip protocol) |
| | blockDAG | Каждая вершина содержит набор транзакций (аналог блока). Каждый блок хеш-ориентирован на несколько родительских блоков |
| | txDAG | Каждая вершина содержит уникальную транзакцию. Ветви содержат непересекающиеся транзакции |

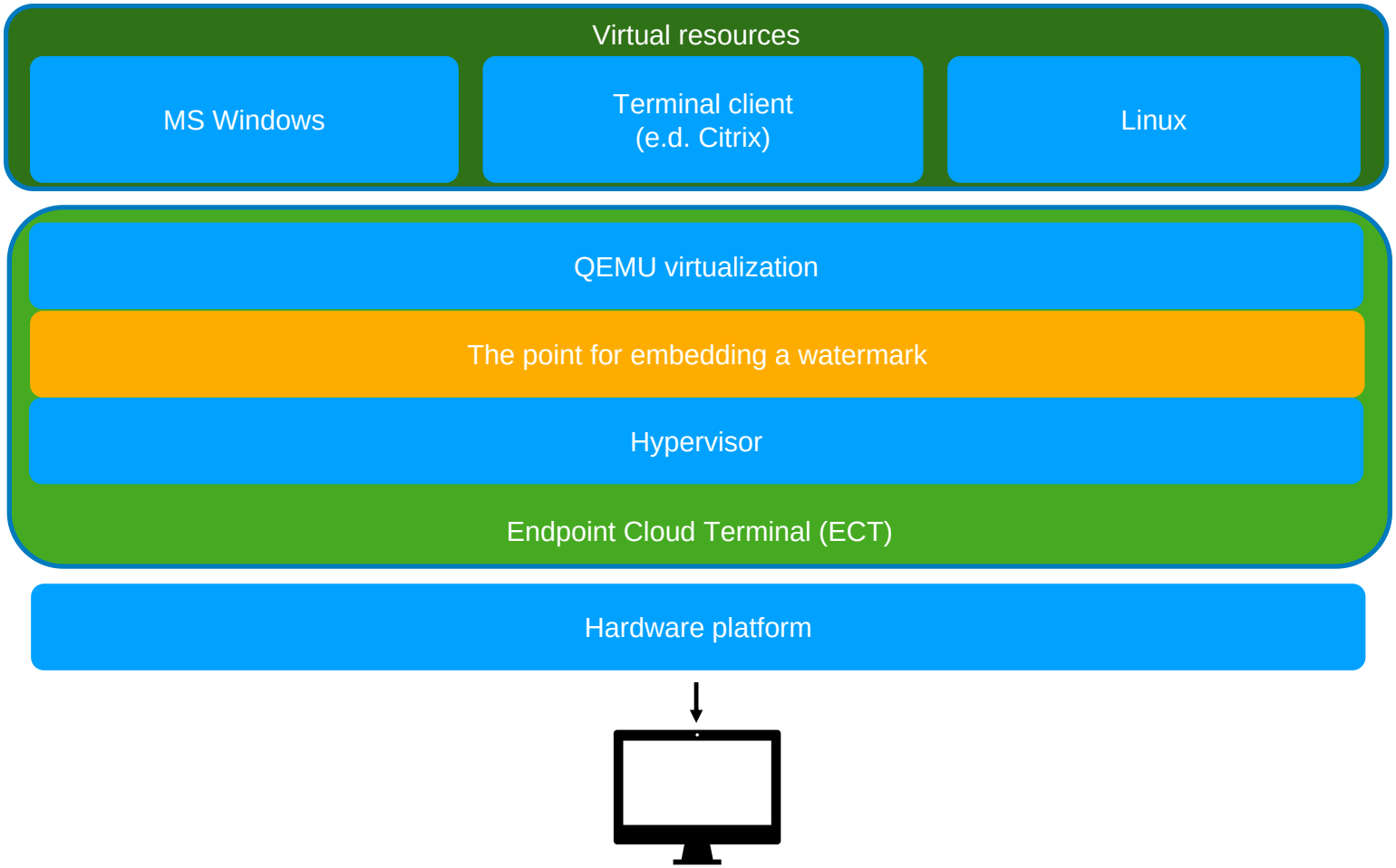
# Модели DLT

| Blockchain | Hashgraph | DAG |
| --- | --- | --- |
|  |  |  |

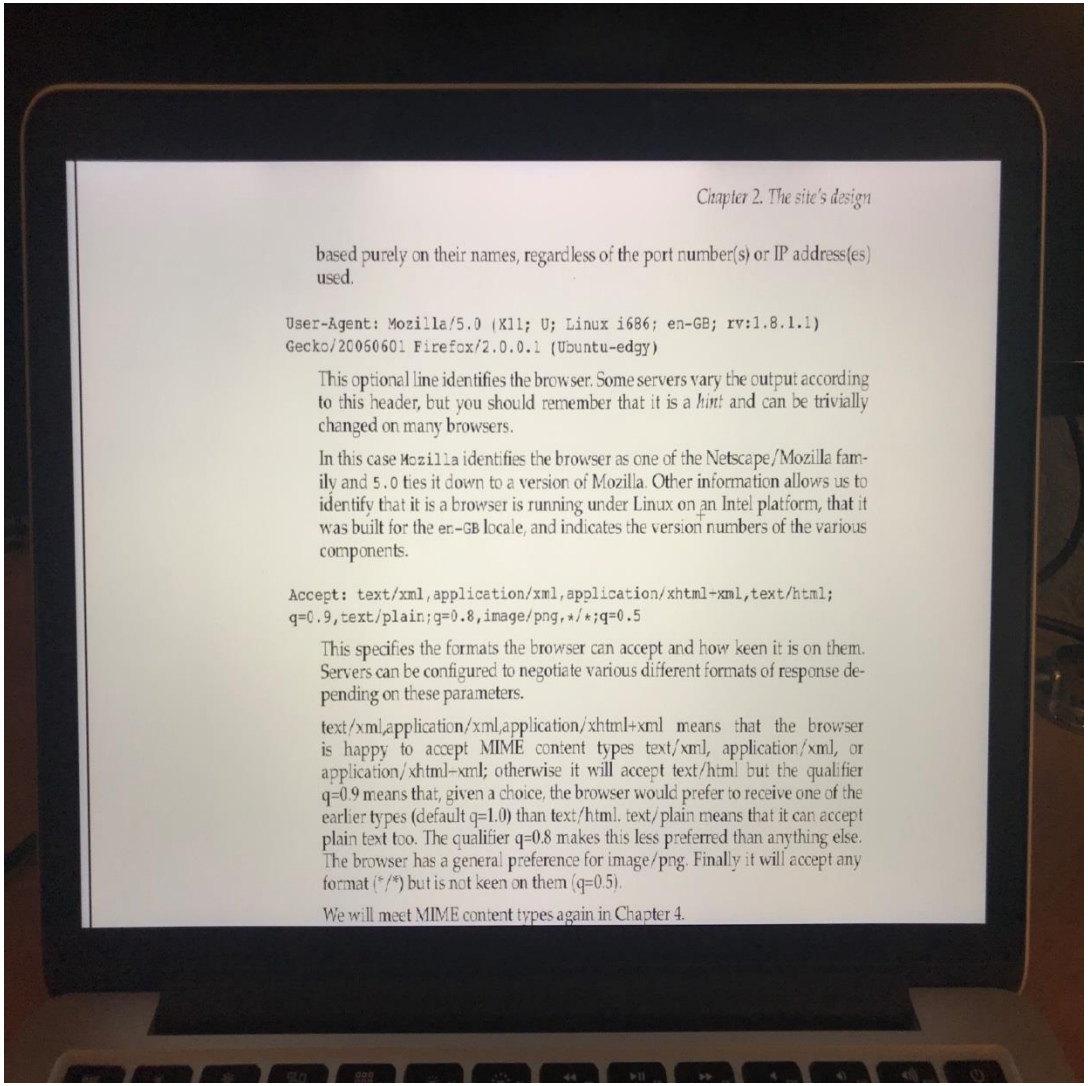
https://cryptor.net/kriptovalyuty/different-types-of-dlts

# Использование цифровых водяных знаков и небайесовские задачи распознавания

# The Research of a Method to Identify a Workplace via a Monitor Snapshot



# Snapshot



*Chapter 2. The site's design*

based purely on their names, regardless of the port number(s) or IP address(es) used.

```
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-GB; rv:1.8.1.1)
Gecko/20060601 Firefox/2.0.0.1 (Ubuntu-edgy)
```

This optional line identifies the browser. Some servers vary the output according to this header, but you should remember that it is a *hint* and can be trivially changed on many browsers.

In this case `Mozilla` identifies the browser as one of the Netscape/Mozilla family and `5.0` ties it down to a version of Mozilla. Other information allows us to identify that it is a browser is running under Linux on an Intel platform, that it was built for the `en-GB` locale, and indicates the version numbers of the various components.

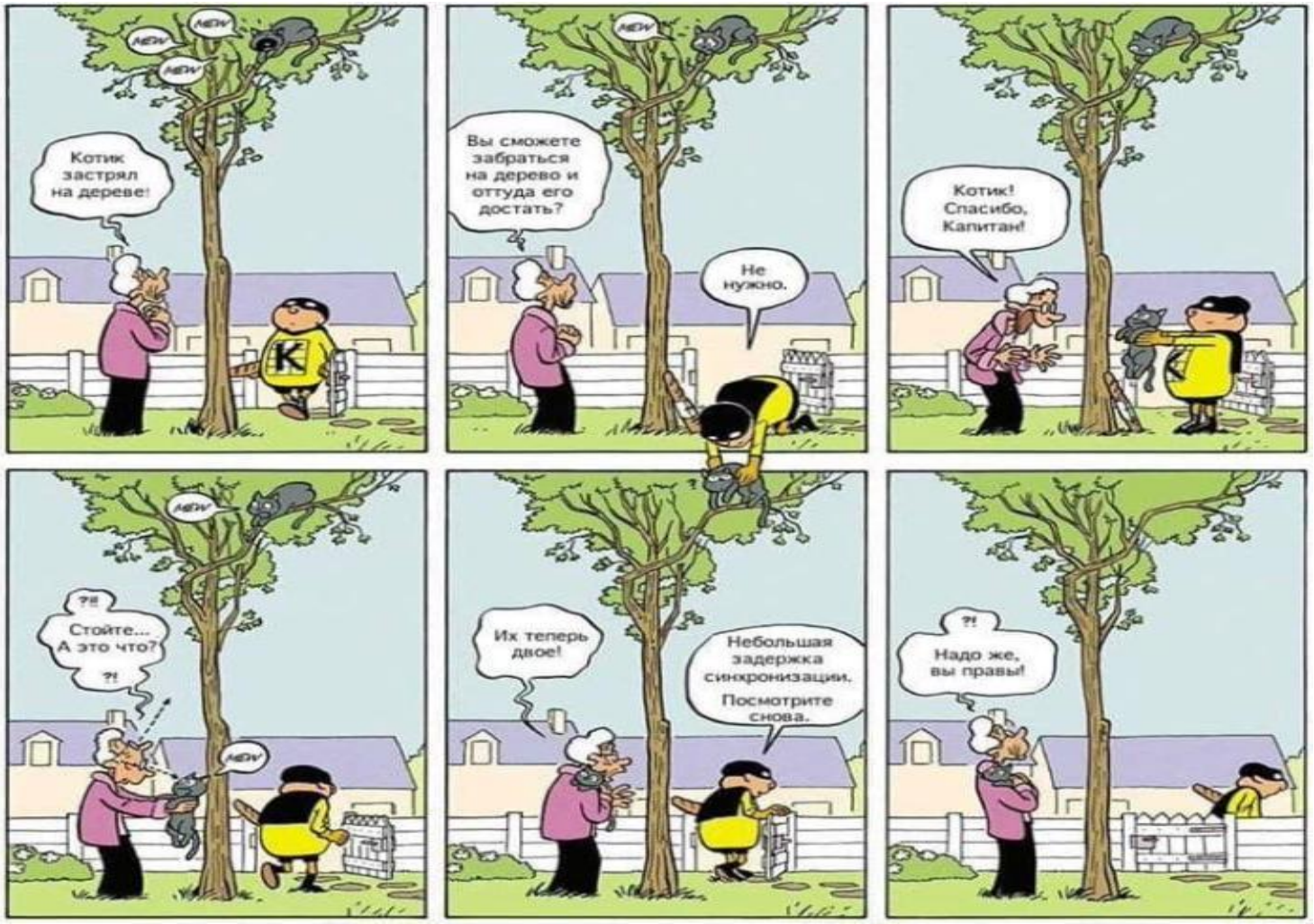
```
Accept: text/xml,application/xml,application/xhtml+xml,text/html;
q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
```

This specifies the formats the browser can accept and how keen it is on them. Servers can be configured to negotiate various different formats of response depending on these parameters.

text/xml,application/xml,application/xhtml+xml means that the browser is happy to accept MIME content types text/xml, application/xml, or application/xhtml+xml; otherwise it will accept text/html but the qualifier q=0.9 means that, given a choice, the browser would prefer to receive one of the earlier types (default q=1.0) than text/html. text/plain means that it can accept plain text too. The qualifier q=0.8 makes this less preferred than anything else. The browser has a general preference for image/png. Finally it will accept any format (*/*) but is not keen on them (q=0.5).

We will meet MIME content types again in Chapter 4.

# Сериализация и проактивная конфигурация сетей


MEW
MEW
MEW
Котик застрял на дереве!
K
MEW
Вы сможете забраться на дерево и оттуда его достать?
Не нужно.
Котик! Спасибо, Капитан!
MEW
?!
Стойте... А это что?
?!
MEW
Их теперь двое!
Небольшая задержка синхронизации.
Посмотрите снова.
?!
Надо же, вы правы!