

2. TCP/IP Networking, Framework & Msg Format

Great Technology For Great Games



DK Moon

dkmoon@ifunfactory.com



Network Protocol

- ✓ 참여자들간 통신이 가능하게 하는 **통신 규약**
- ✓ 구성: Protocol Header + Payload
- ✓ Q) Network Protocol 의 핵심 부분?
- ✓ Header 에 의해 Protocol 이 할 수 있는 것과 할 수 없는 것이 결정됨

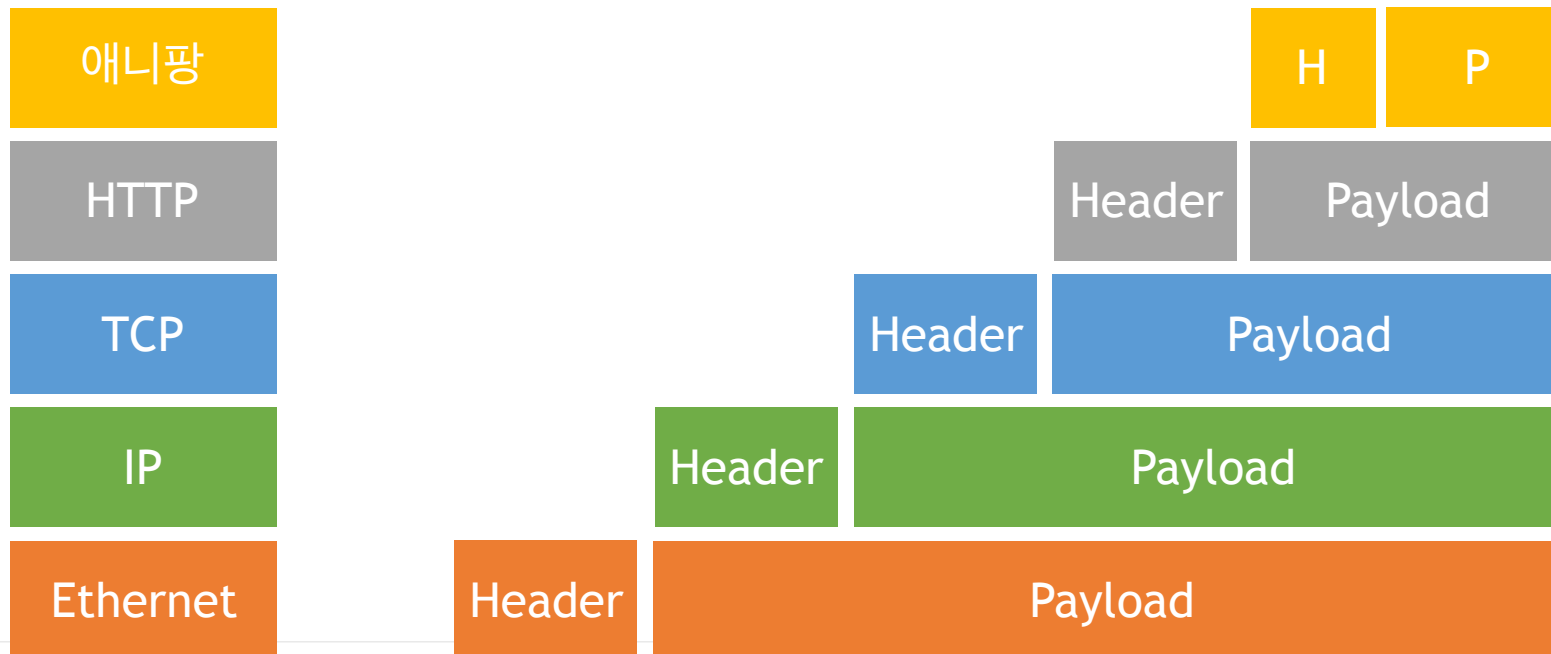
✓ 예) IP Header

| | | | | |
|---|----------------------|-----------------------------|------------------------------|------------------------------|
| Version (4 bits) | IHL (4 bits) | Type of Service (8 bits) | Total Length (16 bits) | |
| Identification (16 bits) | | | Flags (3 bits) | Fragment Offset (13 bits) |
| Time to Live (8 bits) | Protocol (8 bits) | | Header Checksum (16 bits) | |
| Source Address (32 bits) | | | | |
| Destination Address (32 bits) | | | | |
| Options and Padding (multiples of 32 bits) | | | | |



Protocol Layers

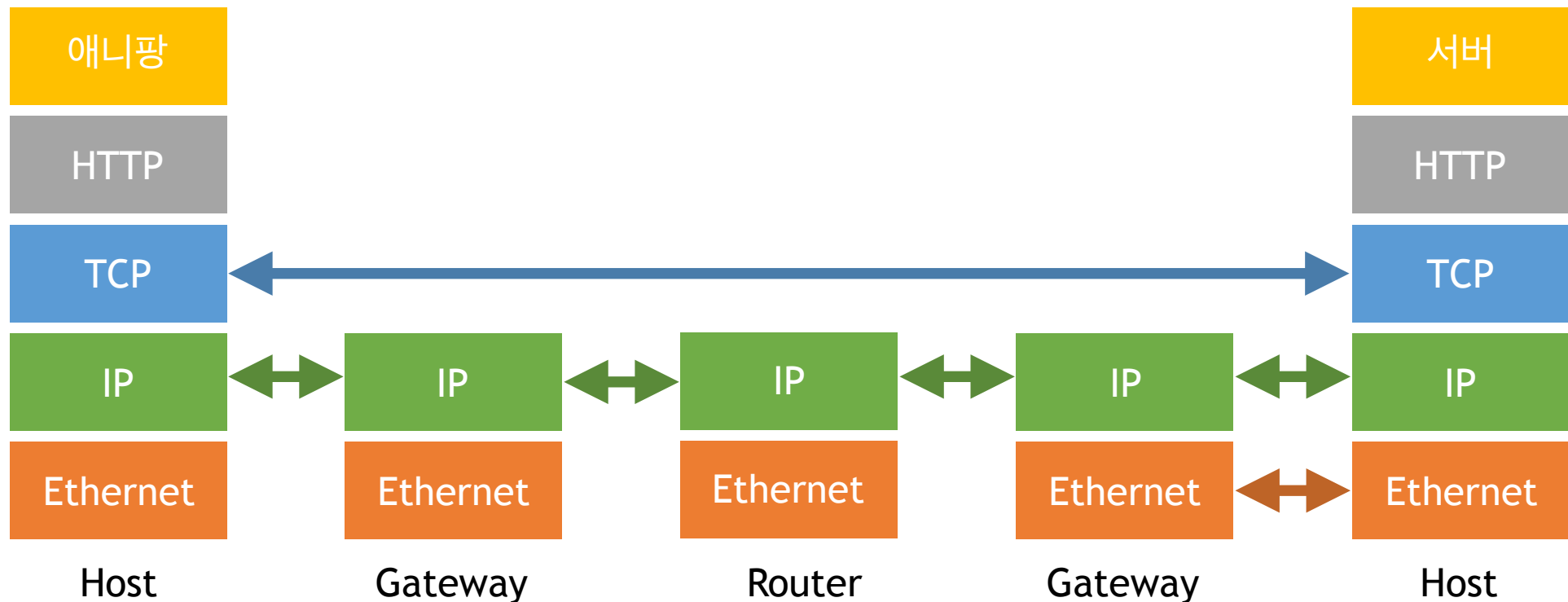
- ✓ stack 형태로 표시함
- ✓ 상위 Protocol 은 하위 Protocol 을 활용해서 (aka. Header 를 채워서), 다른 형태의 서비스 제공
Q) 무엇이 Interface 에 해당하고 무엇이 architecture 에 해당하는가?
- ✓ 상위 Protocol 의 header + payload 가 하위 protocol 의 payload 에 포함됨





Protocol Layers

- ✓ IP: hop-by-hop protocol
- ✓ TCP/UDP: end-to-end protocol





Network Protocol for Gaming

- ✓ IP: Datagram
- ✓ TCP: Stream, 연결 기반, 재전송을 통한 전송 순서 및 송수신 보장
- ✓ UDP: 비연결 기반. IP 의 특성 그대로 Datagram
재전송/순서보장/전송보장이 없으니 빠름. 그러나 유실 가능
- ✓ HTTP(s): TCP 위에서 동작. Request-Reply 방식.
비연결 기반이나 cookie 를 이용한 session 을 통해 가상적인 연결 가능

Socket Programming

<https://goo.gl/cUoLWT>

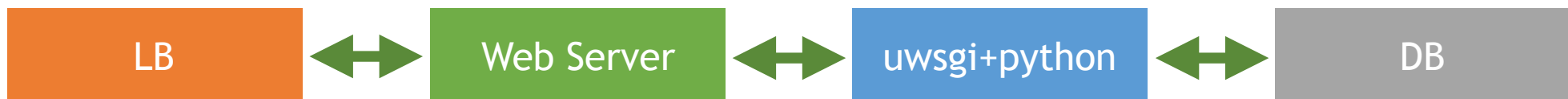
<https://goo.gl/BNuRAD>

Programming 실습: Echo server-client



Python for Game Server

- ✓ Global Interpreter Lock 에 의한 싱글 쓰레드
- ✓ 풍부한 라이브러리 (외부 시스템 연동시 편리)
- ✓ HTTP 서버 모듈을 이용해 외부에서 연결이 들어오는 API 작성 용이
- ✓ 실시간 게임용 서버보다 API 서버로 더 적합
- ✓ 로드 밸런서 + Web Server + uwsgi 를 통한 python worker pool + DB 형태의 시스템 구성





Python Framework

✓ Django

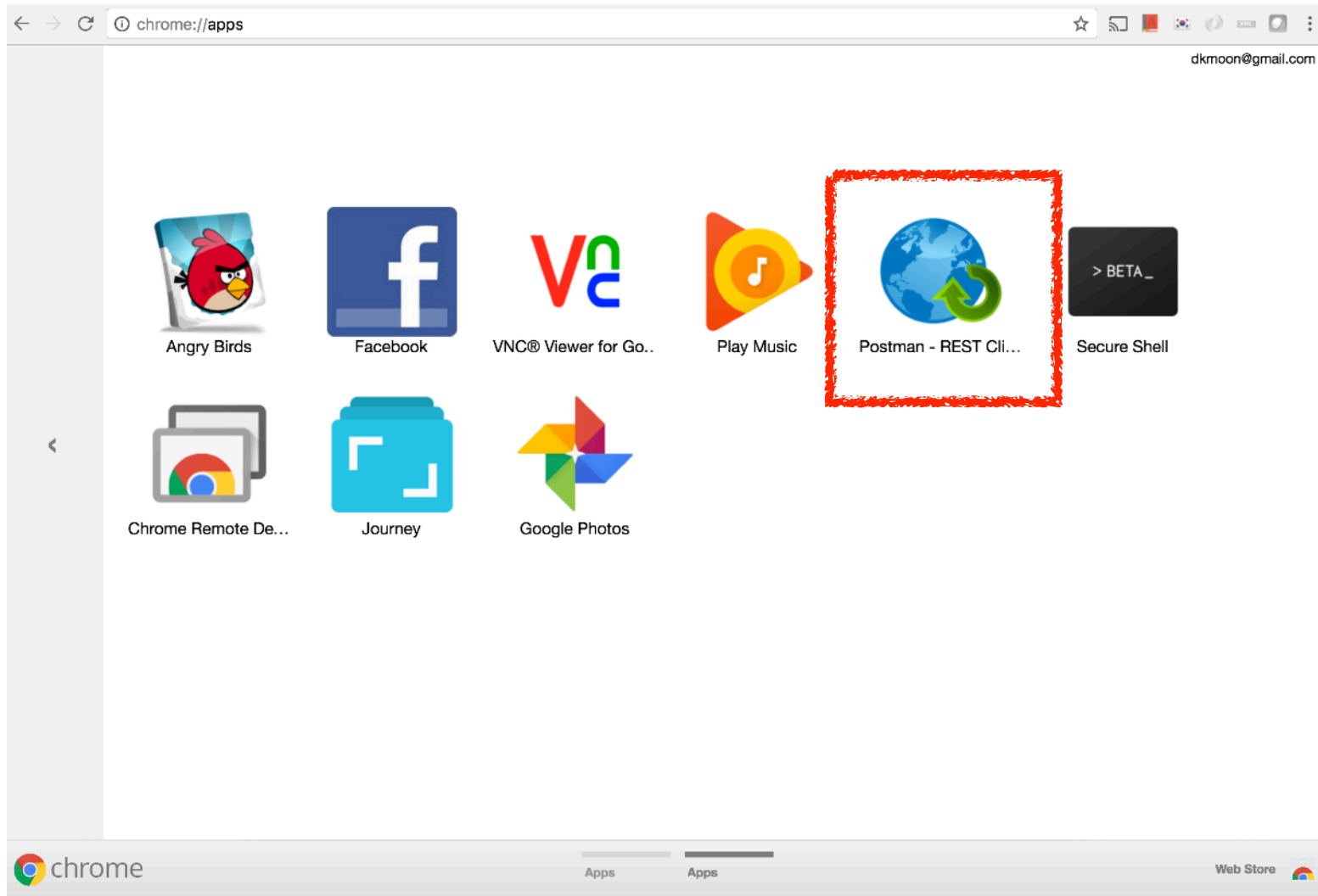
- MVC 패턴에 잘 맞춤
- DB ORM
- API URL => View 매핑
- Template 에 의한 HTML 생성

✓ Flask

- Django 과 유사하나 DB ORM 없음. (SQLAlchemy 활용)



Browse Extension for REST API



Programming 실습: Django API Server

<https://docs.djangoproject.com/en/1.10/intro/tutorial01/>



Django Commands

- ✓ `django-admin startproject hello`
- ✓ `python ./manage.py startapp poll`
- ✓ `python ./manage.py makemigrations`
- ✓ `python ./manage.py syncdb`
- ✓ `python ./manage.py runserver`

```
1 import datetime
2 from django.shortcuts import render
3 from django.views.decorators.csrf import csrf_exempt
4
5 # Create your views here.
6
7 from models import Question
8
9 @csrf_exempt
10 def index(request):
11     if request.method == 'GET':
12         questions = Question.objects.all()
13         print questions
14         return render(request, 'index.html', {'questions': questions})
15     elif request.method == 'POST':
16         q = Question()
17         q.text = request.POST.get('key', 'No key')
18         q.date = datetime.date.today()
19         q.save()
20     return render(request, 'success.html')
```

THANKS!

Great Technology For Great Games, **iFunFactory**



DK Moon



dkmoon@ifunfactory.com



www.ifunfactory.com

