Baidu Smart Game-design, implementation and AI framework

Ping Wu，Architect of Baidu APP

W3C workshop on Web Game

2019

Agenda

1. Introduction
2. Runtime Design and Implementation
3. Developer Support
4. AI framework for the Smart Game
5. Future work

Introduction

1. **Baidu -** the leading Chinese language Internet search provider
2. **Baidu APP –** search and feed twin-engine powered mobile App, more than 180 million daily active users
3. **Baidu Smart Mini Program –** to take advantage of an emerging trend in China where apps with lower frequency are connecting to super apps to bypass the ever-rising cost of app pre-installs

Goals, Definitions and Challenges for Baidu Smart Game

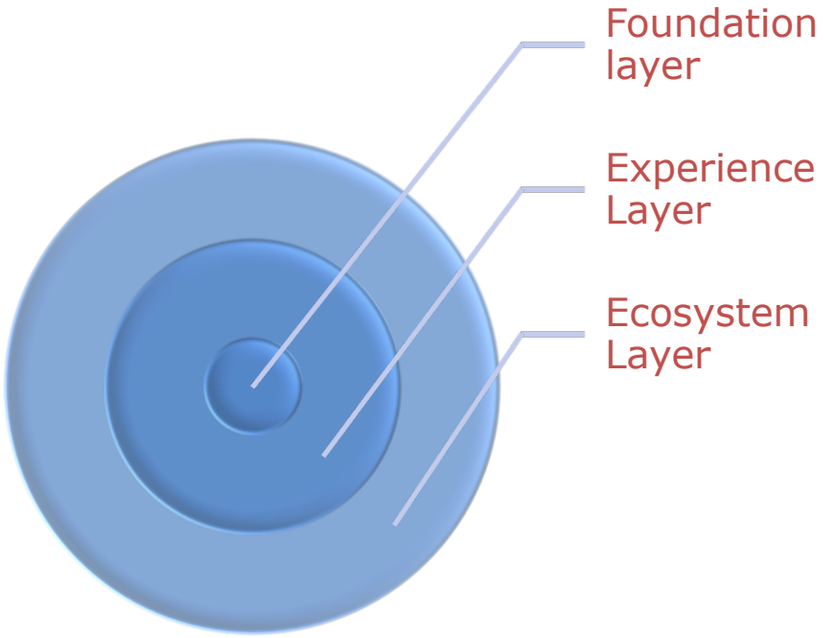
Goals

1. Increase consumable resource type for Baidu App, set up game scenario to improve increase in DAU，user time，stickiness
2. Set up mini game ecosystem, explore distribution and monetization patterns

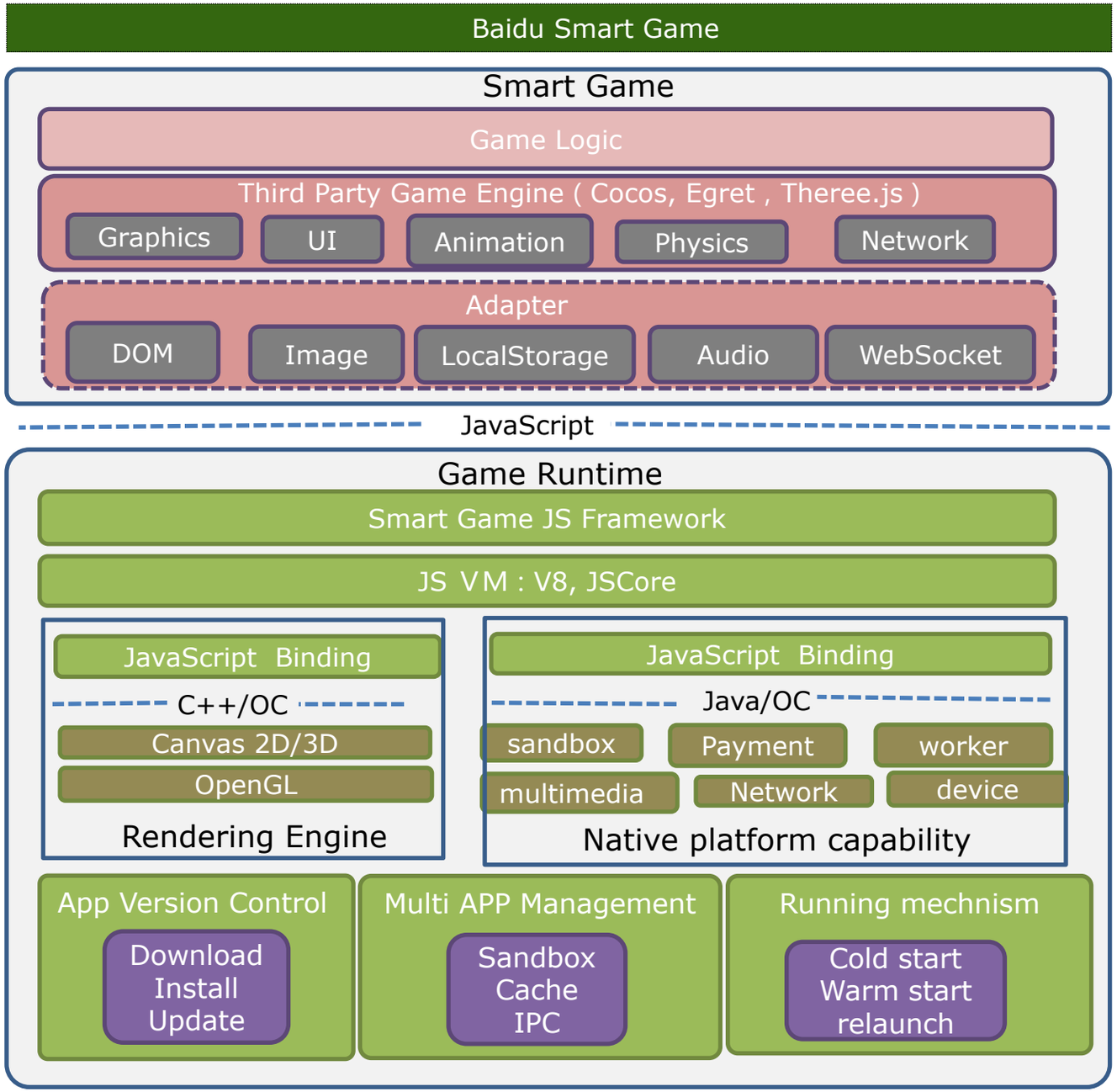
Definitions

1. Smart game is runnable in a browser-like environment built-up by super app, which is instant play without installation
2. A super container for forms like web game, smart mini game and cloud game
3. Include light, medium and serious game types

- Technical challenges for Smart Game
  - **Foundation layer**
    - Runtime architecture and implementation
  - **Experience layer**
    - Core user experience-high performance, low latency, high-availability
  - **Ecosystem layer**
    - Easy-to-develop, scalable game importing and rich distributable game types



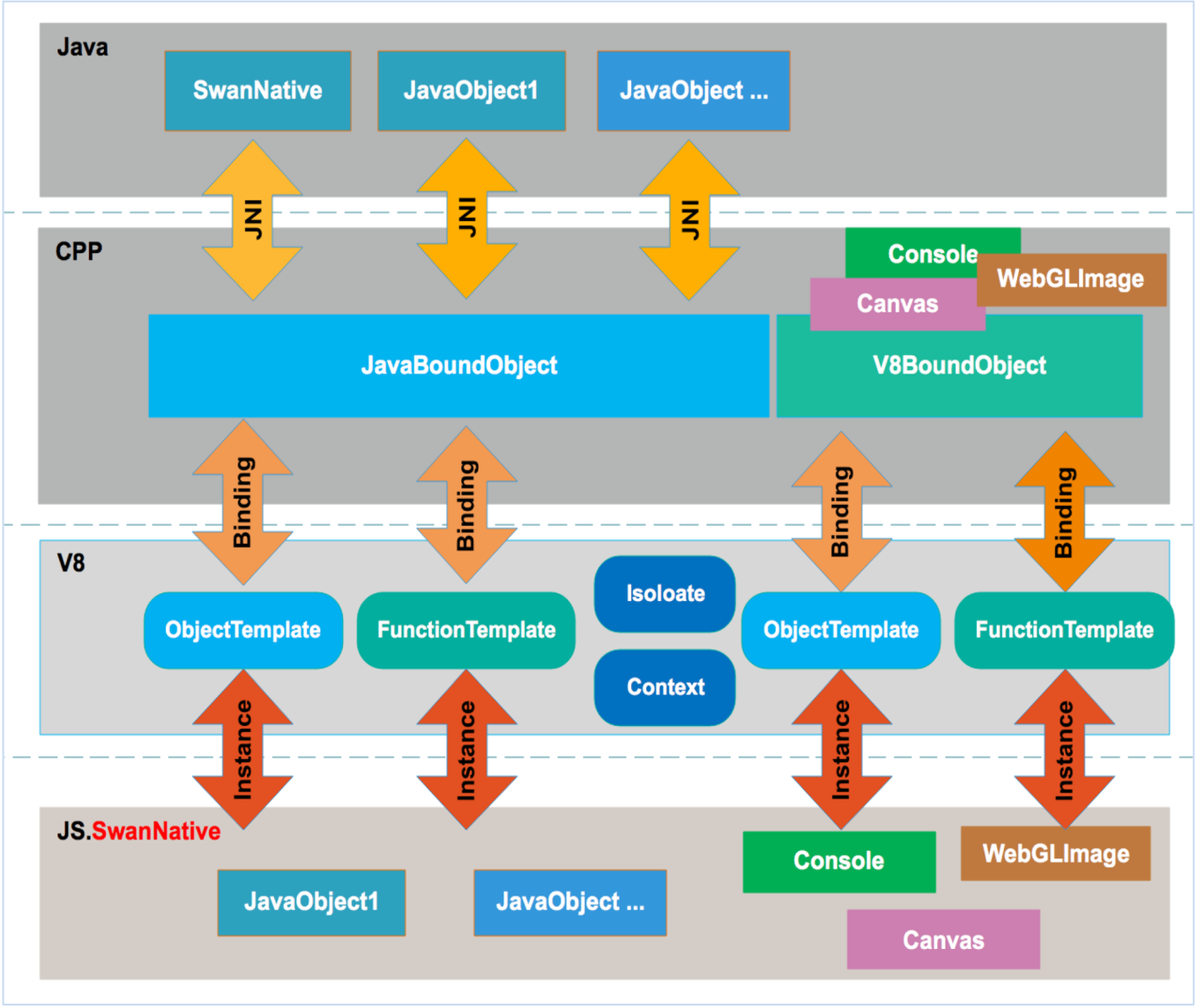
Smart Game runtime design



Runtime Kernel :

- Construct a pure JS execution and native rendering environment with canvas 2D and 3D implementation.
- Provide an app model abstract for web game, like a web app package, lifecycle, version control, multi-app management.
- Provide rich native APIs in Baidu APP by JS binding framework.

JSB binding Framework- general and efficient JS-Native operation



Design principal:

- efficient
- general

Key elements:

- object model
- Life cycle
- Memory management

JSB binding framework- 10-200x improvement compared with JS bridge scheme

Loop 1000 for a simple function

```
// Javascript impl
function testTimeFunc() {
  var x = 0;
  x = x + 2 + x * 6;
}

swanNative.console.time();
for (var i = 0 ; i < 1000; i++) {
    testTimeFunc();
}
swanNative.console.timeEnd();
```

	JS	JSB JAVA/OC impementation	JSB C++/C implementation	JS bridge with scheme
iOS	1.75 ms	0.65 ms	0.197ms	2.15ms
android	1ms	0.82ms	0.05ms	NA

local data passing 1MB

```
function request(){
    swan.request({
        url: 'http://xxxxxxxx',
        method: 'GET',
        responseType: 'text',
        dataType: 'json',
        success: res => {
        }
    });
}
```

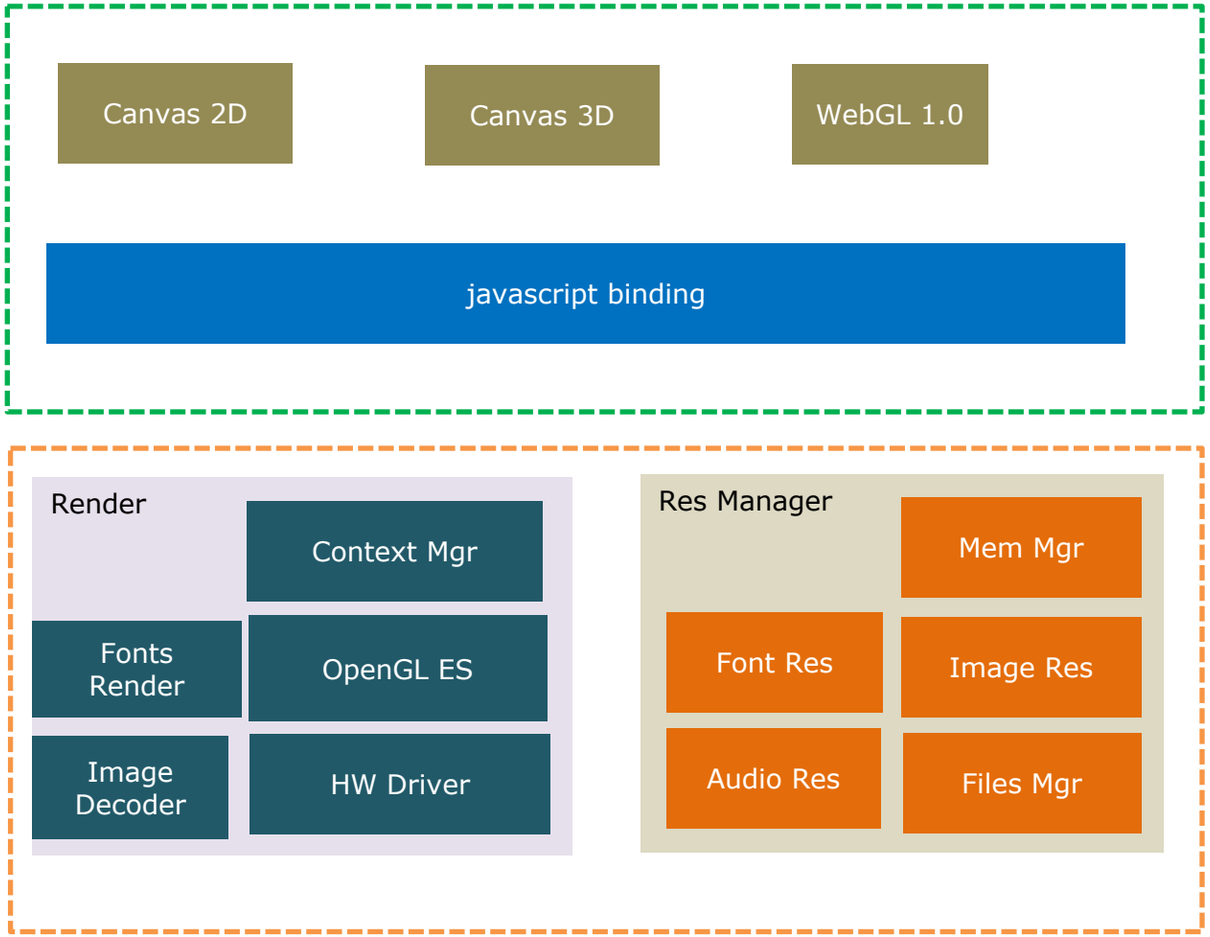
	JSB text	JSB arrayBuffer	Scheme with text	Scheme with arrayBuffer
iOS	11.5ms	3.5ms	795ms	1172ms

Efficient JS execution engine

- **V8**
  - Minimize parse and compile time
  - Code Cache
  - GC Incremental marking
  - GC at each game over

initialization stage optimization by v8

Lightweight Native Rendering Engine



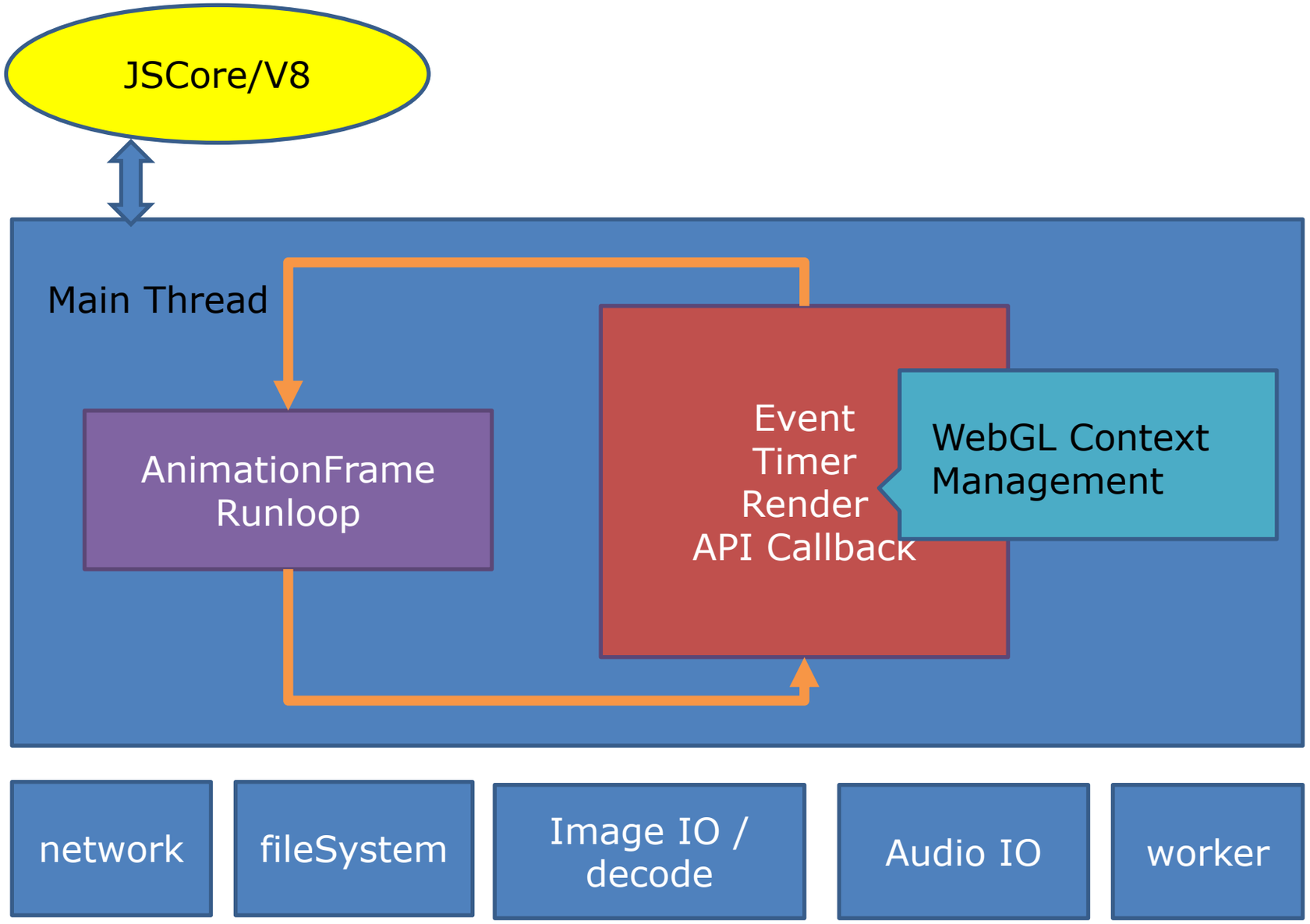
Design key elements:

- A lightweight standard implementation of HTML5 canvas 2D and 3D features in webGL 1.0/openGL 2.0
- Fonts render with FreeType/CoreText
- Image decoder

Key problems:

- **Context Switch**
  - Minimize context switch frequency and must-to-update switching states by diff
- **texSubImage(canvas,...)**
  - Optimize CPU data copy with CopyTexImage2D
- **Font Render**
  - Arrange memory areas based on font size

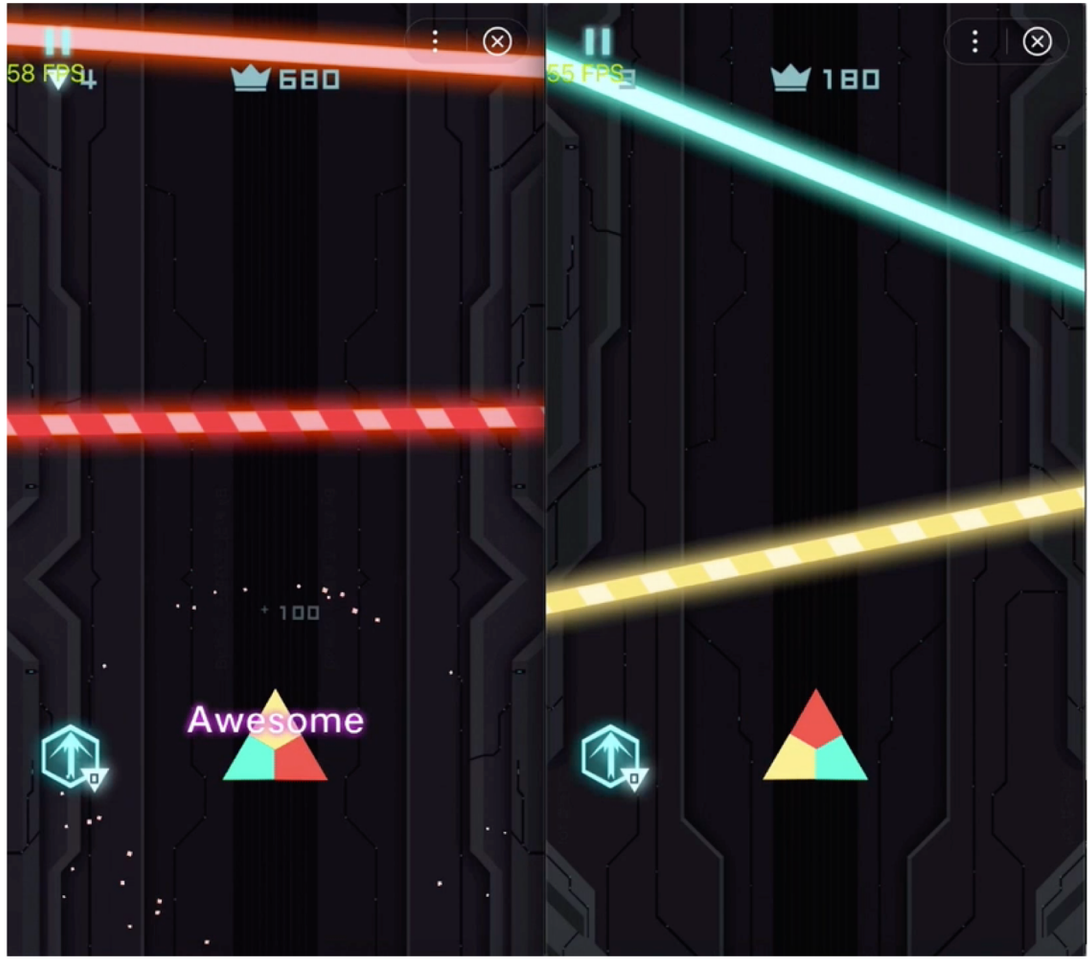
Threading model design in native rendering environment



Design key points:

- main thread for JS execution and rendering engine.
- Parallel image decoder, image IO, audio IO from render thread.

Key modules for Smart Game –file system, audio and network



sprint stage audio optimization

- **Audio**
  - Optimize cache implementation
  - JSBinding
- **File system**
  - Optimize file IO
  - JSBinding
- **Network**
  - multi-threaded download
  - Background download

Benchmark for Smart Game Runtime

Micro benchmark cases

Image | Render | FS

Real world cases

Laya | Aquarium | Box2D

Benchmark Framework

Cases Manager | Cases Runner | Performance Manager

Baidu Smart Game Runtime

Benchmark Glue | Image | File System | JSEngine | Render

UI Framework

Pixi.js | Gown.js

Benchmark - Dashboard

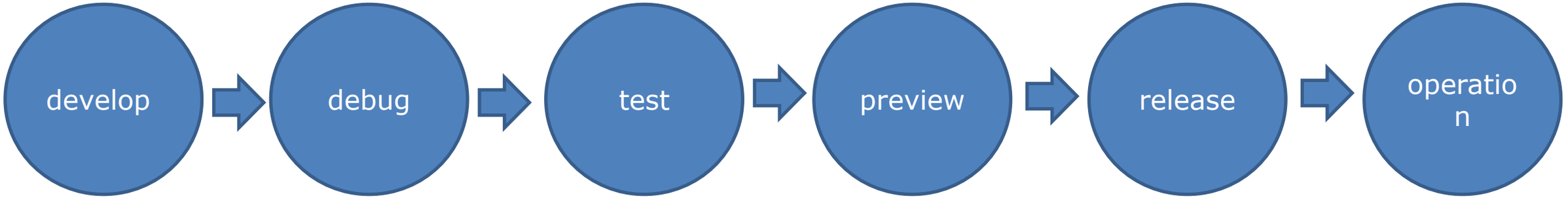
```ts
interface _naSwan.Performance {
    double now();
    // 返回当前的 CPU 占用
    float getCPU();
    // 返回当前的进程内存
    int getMemory();

    // 注册 GC 回调函数
    void registerGCCallback(Function);
    /* 以下接口目前android独有 */
    // 详细的内存信息
    MemoryInfo getMemoryInfo();
    // 设置、进程、线程CPU时间变化量
    CpuUsageTime getCpuUsageTime();
    // V8虚拟机的堆内存信息
    HeapStatistics getV8HeapStatistics();
    HeapSpaceStatistics getV8HeapSpaceStatistics();
}
```

Smart Game Benchmark

Baidu APP scan code

Developer Support for Productivity



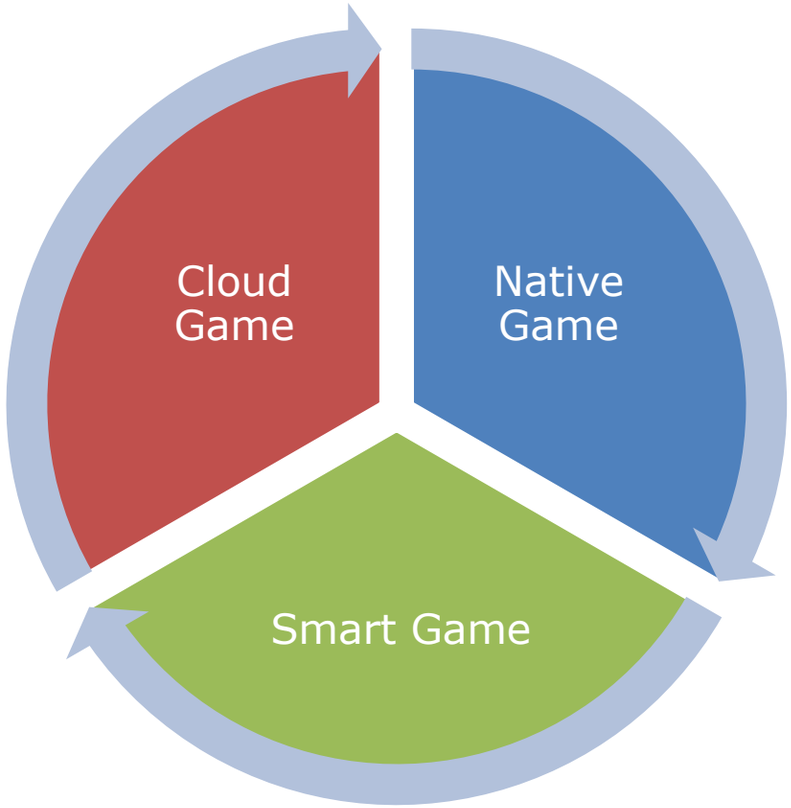
- 60,000 smart program developers
- 150,000 smart programs created
- 20,000 smart programs online
- 100 million DAU

Scalable large-scale game import with main stream game engines

- **Cocos** :http://docs.cocos.com/creator/manual/zh/publish/publish-baidugame.html
- **Egret:** http://developer.egret.com/cn/github/egret-docs/Engine2D/minigamebaidu/getStart/index.html
- **Laya**
- **Unity**

Super container for Cloud Game, Smart Game and Native Game

Cloud game: based on video stream or gl commands stream



Veloce: Instant APP invocation

WASM:Web-Assembly version

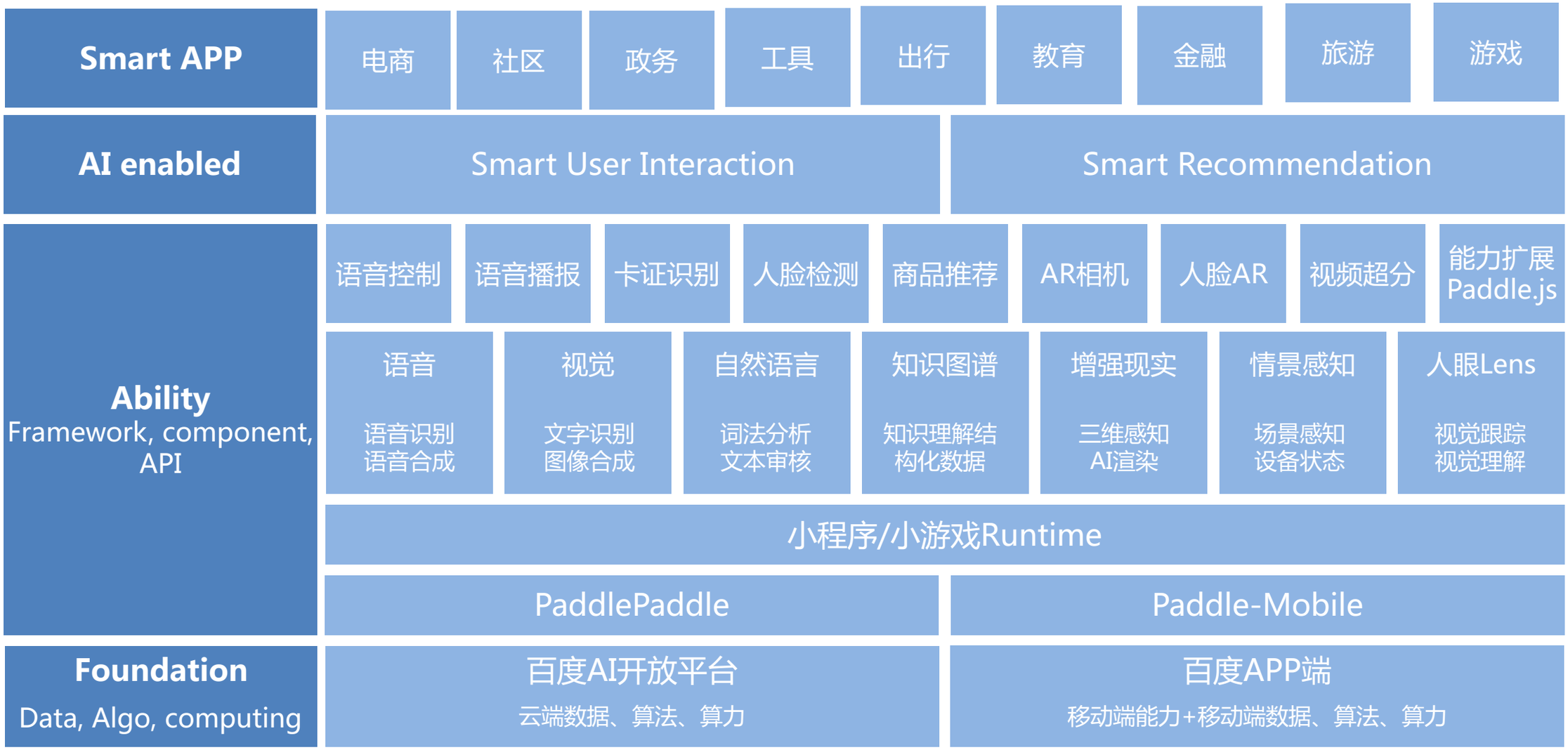
Smart Game : based on pure JS execution environment and native rendering.

AI Framework for Baidu Smart Game

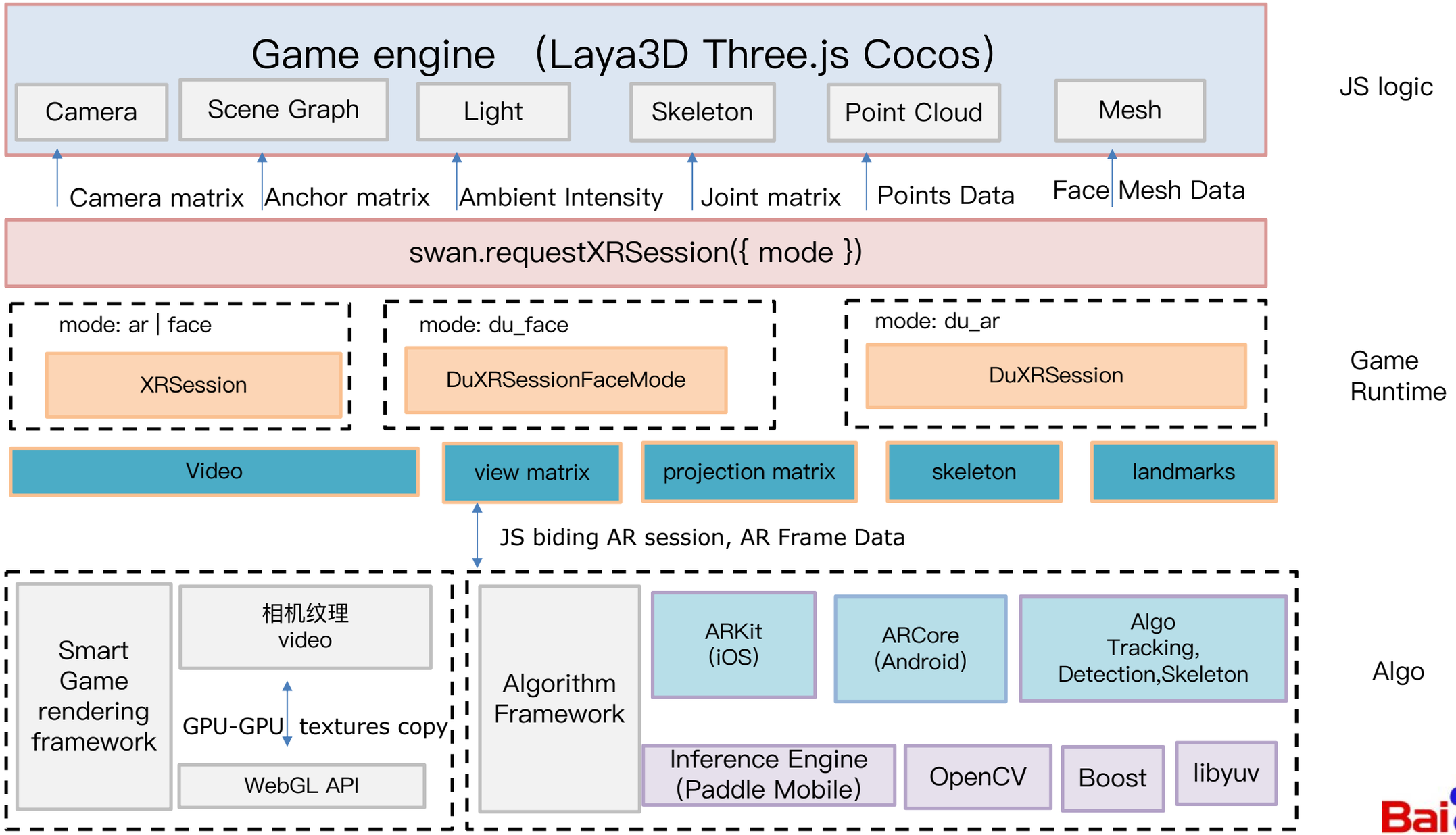
Challenges for AI enabled Smart Program and Game

- Rich and easy-to-use AI capabilities
- Same user experience as native app
- Flexible and general programmable abstraction
- Customized AI capabilities and application
- Framework for vertical industry and user scene

AI framework for Baidu Smart Program and Game



Architecture for Smart Game AR and Face



Design & Implementation rules for AR and Face

- **JS Framework** : compatible with WebXR , ARCore , ARKit API definitions, abstract developer-friendly JS object like Session , Frame, Video
- **Game runtime** : Communicate efficiently between JS and algorithm using JSB binding framework. Render the camera texture and canvas texture in the same hybrid rendering layer.
- **Algorithm**: Utilize AR ability of the system ARKit and ARCore and Baidu DuMix AR platform and Face Algo as a supplement

AR API

1. init Session： swan.requestXRSession({ mode: ‘ar’ })

   异步接口，初始化成功进入 success 回调，失败进入 fail 回调

2. getFrameData：xrSession.getFrame()

   - getViewMatrix(), getProjectionMatrix()

     三维相机的变换矩阵，可以直接用于 THREE.js 等 3D 引擎

   - getPointCloud()

     特征点点云，包含点云的(x, y, z) 的三维坐标

   - getTrackablePlanes()

     获取检测到的空间中的平面

   - hitTest(x, y)

     根据屏幕像素坐标得到三维空间坐标

   - createAnchor(poseMatrix)

     创建锚点，锚点是三维空间中一个固定的位置和朝向

   - getLightEstimate()

     环境光强度

3. close Session： xrSession.end()

ARCore

Face API

1. Init Session： swan.requestXRSession({ mode: ‘du_face’ })

   mode: ‘du_face’ 初始化人脸 AR 模式，未来会支持更多的模式去支持更多的 AR 场景（人体骨骼，物体追踪，图像追踪）

2. getFrameData：xrSession.getFrame()

   - getCameraVideo()

     相机 Video 对象，可以直接用于 THREE.js 等引擎作为纹理数据

   - getUpdatedTrackableFaces()

     当前帧检测到的人脸

   - face.poseMatrix

     人脸姿态的 4x4 变换矩阵

   - face.landmarks

     人脸特征点归一后的 (x, y) 坐标

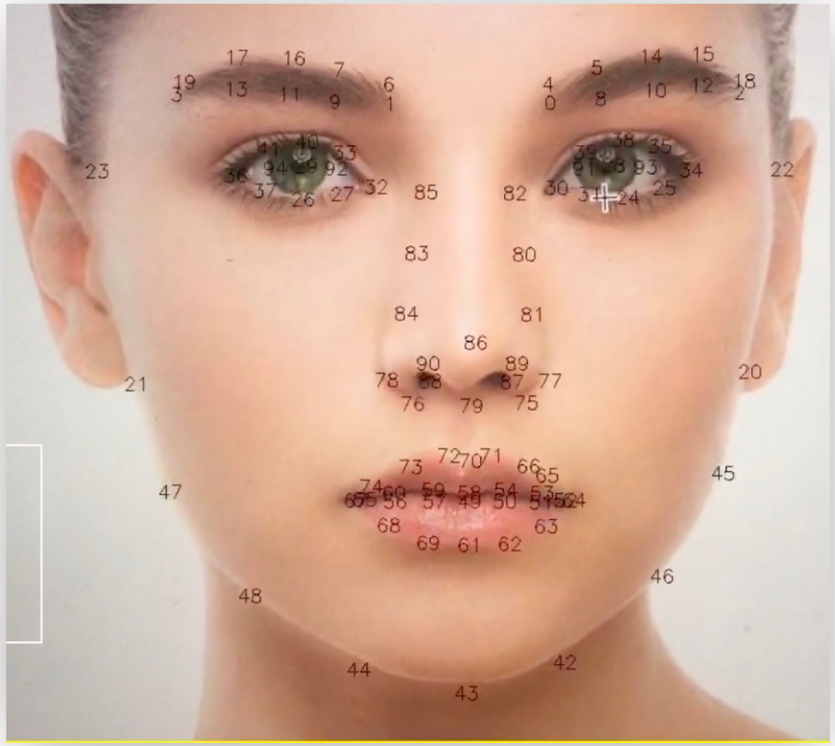
   - face.skeleton

     人脸骨骼节点的变换矩阵

   - face.blendShapes

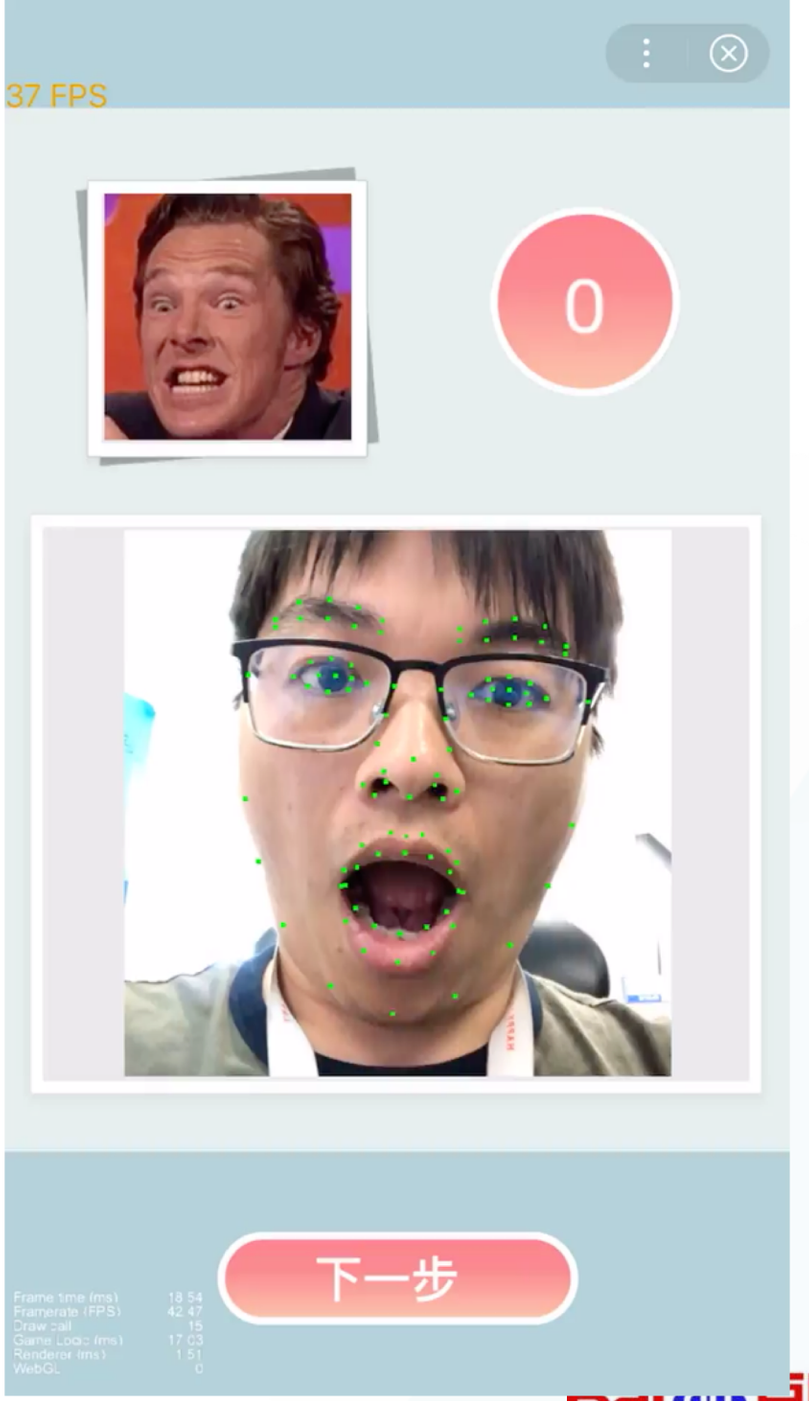
     表情系数

3. close Session： xrSession.end()



```
cc.Class({
    onLoad() {
        swan.requestXRSession({
            mode: 'du_face',
            drawCameraBackground: false,
            success: xrSession => {
                this._cameraVideo = xrSession.getCameraVideo();
            }
        });
    },

    update() {
        this.cameraTexture.update({
            image: this._cameraVideo
        });
    }
});
```



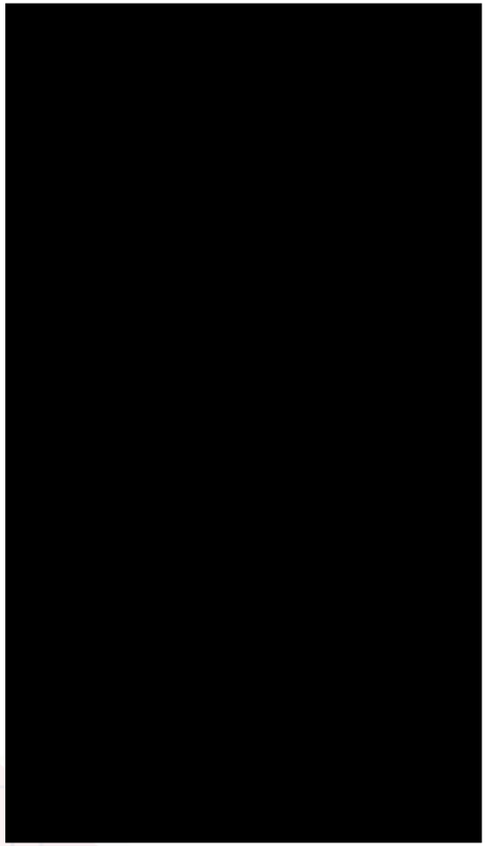
```
const frame = this._xrSession.getFrame();
const face = frame.getUpdatedTrackableFaces()[0];
if (face) {
    const videoBlendShapes = face.blendShapes;

    const a = videoBlendShapes.map(bs => bs.weight);
    const b = refBlendShapes.map(bs => bs.weight);

    this.totalScore.string = (similarity(a, b) * 100).toFixed(0);
}
```

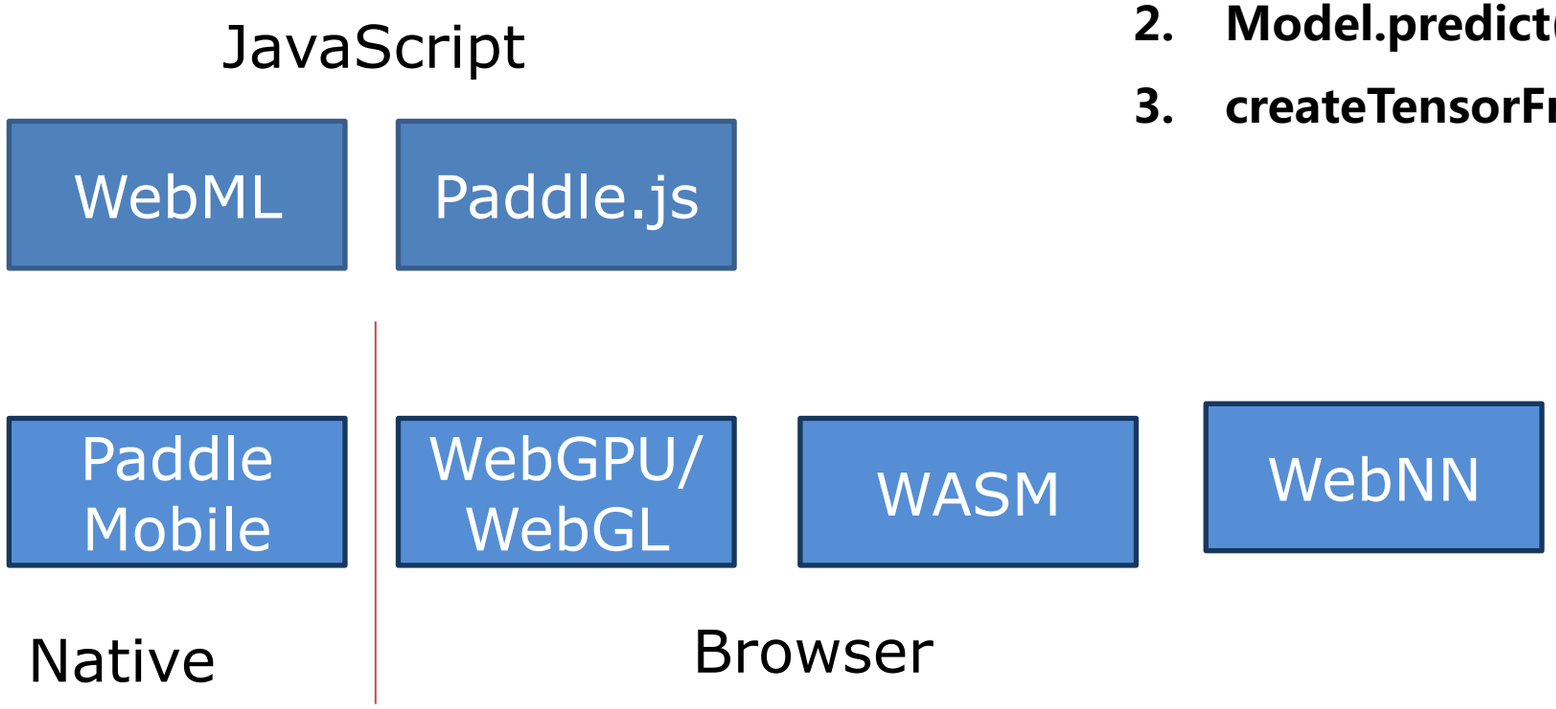
```
const frame = xrSession.getFrame();
const face = frame.getUpdatedTrackableFaces()[0];
if (face) {
    const skeleton = face.skeleton;
    updateMeshSkeletons(skeleton);
}
```

Face Game Demo



AR Demo

WebML & Paddle.js-customize your own AI application



1. **loadModel(url | id) => Model**
2. **Model.predict(Tensor) => Tensor**
3. **createTensorFromImage(Image) => Tensor**

Future Work

1. Smart Game is a very promising web tech powered by native experience
2. Provide more building blocks, more AI enabling features, more prosperous game society

Thanks ！