XII. Architectures and Technologies for Mobile Application Development

Prof. Tsvetozar Georgiev
University of Ruse

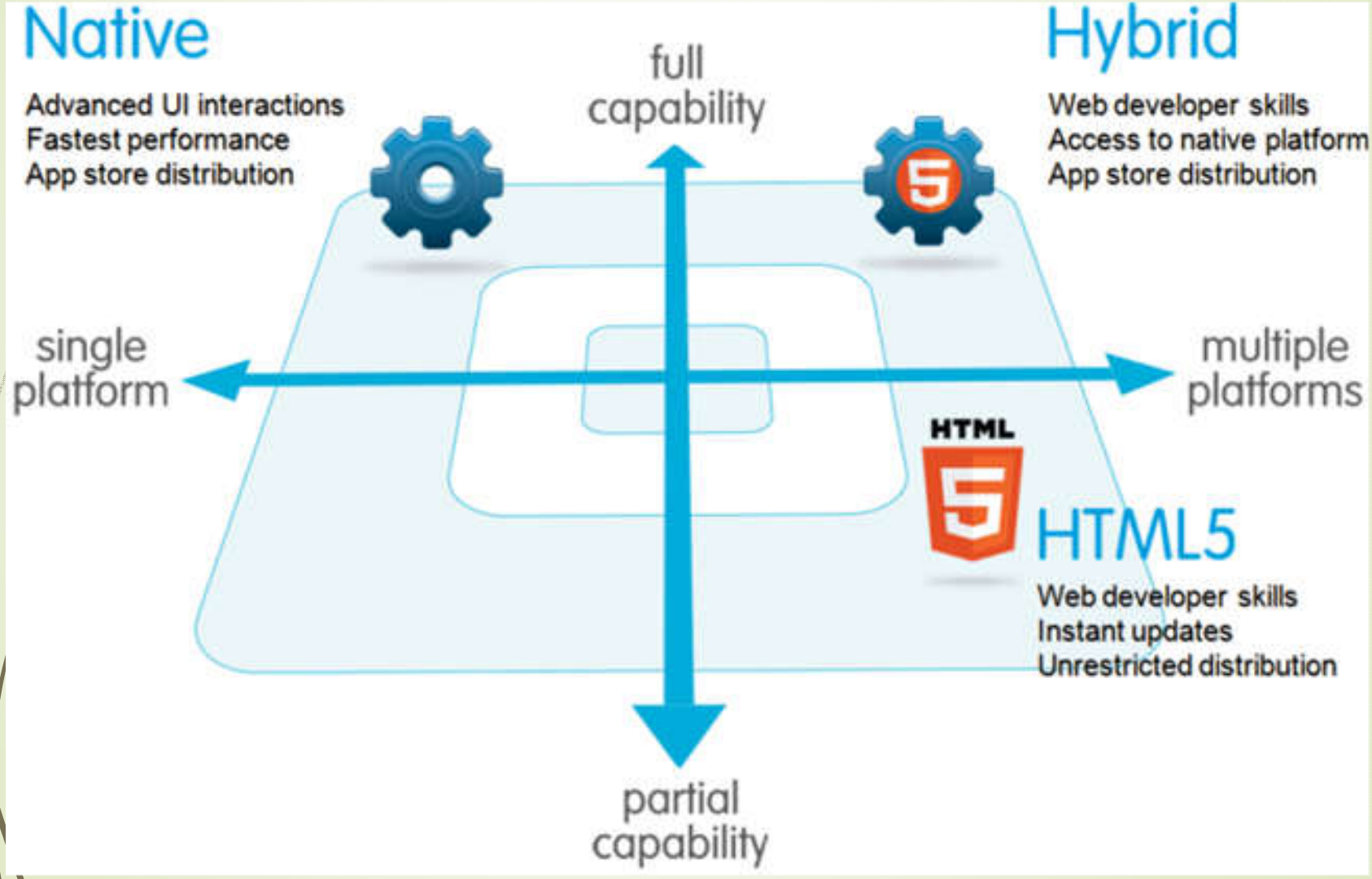
Types of architectures

- Currently, there are three main types of architectures for mobile applications – thin client, thick client and smart client.
- To determine which of these architectures is most suitable for developing a given mobile application, the following questions must be answered:
  - Who are the end users of the application?
  - Do users have special requirements for the device or will the application determine its appearance?
  - What is the main purpose of the app?

Types of architectures

- Is constant access to databases required?
- Does the application require a wireless connection and what type?
- How will the application be used?
- How will the application be distributed and updated?

- The choice of means for its development depends on the architecture of the mobile application.


Native
Advanced UI interactions
Fastest performance
App store distribution
full capability
Hybrid
Web developer skills
Access to native platform
App store distribution
single platform
multiple platforms
HTML
HTML5
Web developer skills
Instant updates
Unrestricted distribution
partial capability

Types of architectures

1. Thin client

- The term "thin client" refers to an Internet-based application that is executed by the mobile browser of the client device.
- This means that the client device does not need to install any software other than a web browser.
- The thin client should be completely independent of a specific browser or operating system.


YOUR APP
WEB UI & LOGIC
Mobile UI Components
Develop mobile web applications in the familiar force.com environment on a proven architecture stack
Visualforce
Develop mobile web applications in the familiar force.com environment on a proven architecture stack
JavaScript Remoting
Invoke Apex controller methods directly from a mobile application for optimized performance
Static Resources
Include mobile optimized third parties frameworks like JQuery Mobile, iScroll, and Sencha
HTML
5

Types of architectures

1. Thin client. Advantages and disadvantages

- This type of mobile application has the following advantages:
  - faster development compared to "thick client" applications;
  - no software installation is required in the memory of the user's mobile device;
  - the use of a mobile web browser to play the application eliminates potential problems of incompatibility with some operating systems or application software, providing an opportunity to develop a single application;

Types of architectures

1. Thin client. Advantages and disadvantages

- access from anywhere and at any time, which ensures wider distribution;
- lower cost to maintain the application;
- possibility to easily update the content;

- Disadvantages of thin client applications are:
  - lower degree of interactivity compared to thick clients, since not all mobile web browsers support JavaScript to the same extent;
  - limitations on the size of media files that are transferred over the wireless Internet;

Types of architectures

1. Thin client. Advantages and disadvantages

- due to the lower degree of interactivity, these applications require information to be presented in small portions, on more screens, resulting in more user actions, requests to the server, and slower performance compared to thick clients;
- due to the continuous emergence of new mobile web browsers and the rapid change of technology, it is difficult for thin clients to take advantage of their best capabilities. To ensure compatibility, developers are forced to accommodate the capabilities of older browsers;

Types of architectures

1. Thin client. Advantages and disadvantages

- users cannot develop offline with the application and therefore a continuous wireless Internet connection is required;
- an increasing variety of mobile web browsers with different functional capabilities;
- possible need to identify the type of mobile device and to adapt the application;
- limited access to the specific resources of the mobile device due to security reasons.

Types of architectures

2. Thick client

- The term "thick client" refers to an application that is loaded into the memory of the user's mobile device.
- It can communicate with a server using a wireless or wired connection to synchronize data.


YOUR APP
NATIVE UI & LOGIC
OAuth2
Secure authentication and refresh token management
API Wrappers
Interact with Salesforce REST APIs with popular mobile platform languages
Secure Offline Storage
Store business data on a device with enterprise-class security
Push Notifications
Dispatch real-time alerts directly to mobile devices

Types of architectures

2. Thick client. Advantages and disadvantages

- This type of mobile application has the following advantages:
  - have more media playback capabilities, a better user interface, and a greater degree of interactivity than thin clients;
  - can store relatively large volumes of data directly on the mobile device, which speeds up access to them;
  - run faster and do not require a continuous connection to a server;

Types of architectures

2. Thick client. Advantages and disadvantages

- can directly interact with the resources of the mobile device, receiving useful information - location, contact list, acceleration information, etc.;
- development environments are improving very quickly;
- information can be obtained on how users interact with the application.

Types of architectures

2. Thick client. Advantages and disadvantages

- Disadvantages of fat client are:
  - greater equipment and development costs if the application must support different platforms;
  - difficult updating, adding new functionality and longer debugging time, as it requires installing the software in the memory of the mobile device;
  - there are currently many development environments and one should choose the one that best suits the needs;

Types of architectures

2. Thick client. Advantages and disadvantages

- developing rich applications requires more resources than developing a mobile website;
- difficult portability of the application to another type of mobile device and/or another operating system, which most often leads to the need to develop a different version for each mobile platform.

Types of architectures

3. Smart client

- This term refers to an application that uses local resources on a mobile device, is based on web services, and can be distributed and updated from a centralized server.
- These applications can work both wirelessly connected to a server and without such a connection.
- They combine the capabilities of thin and fat customers.

Visualforce

Develop mobile web apps in the familiar Force.com environment on a proven architecture stack

JavaScript Remoting

Invoke Apex controller methods directly from a mobile application for optimized performance

Mobile Components

Reusable Visualforce-based building blocks for constructing mobile apps through customizable components

Container

Embed HTML5 apps inside a container to access powerful native device functionality

Types of architectures

3 . Smart client. Advantages and disadvantages

- The advantages of smart clients are:
  - ability to distribute and update from a server via the Internet;
  - are less dependent on the platform on which they work, as they are based on web services;
  - greater interactivity and faster completion of tasks, as with fat clients;
  - some of the platforms have built-in security and management capabilities;
  - some of the platforms provide use of built-in resources for communication (e-mail, chat, etc.).

Types of architectures

3 . Smart client. Advantages and disadvantages

- Disadvantages of smart clients are:
  - there is no standardization of the APIs used;
  - higher price, because unlike platforms for developing "thick clients", those for developing applications of the "smart client" type are not free;
  - vary in their "thickness" depending on the functional capabilities of the mobile device.

Modern technologies for rapid development

- Mobile platforms are the fastest growing, and as a result, many people and businesses are turning to mobile apps to engage their audience more effectively.
- However, this is not an easy task, as creating mobile applications requires a good command of modern programming languages, such as Objective-C (iOS) or Java (Android).

Modern technologies for rapid development

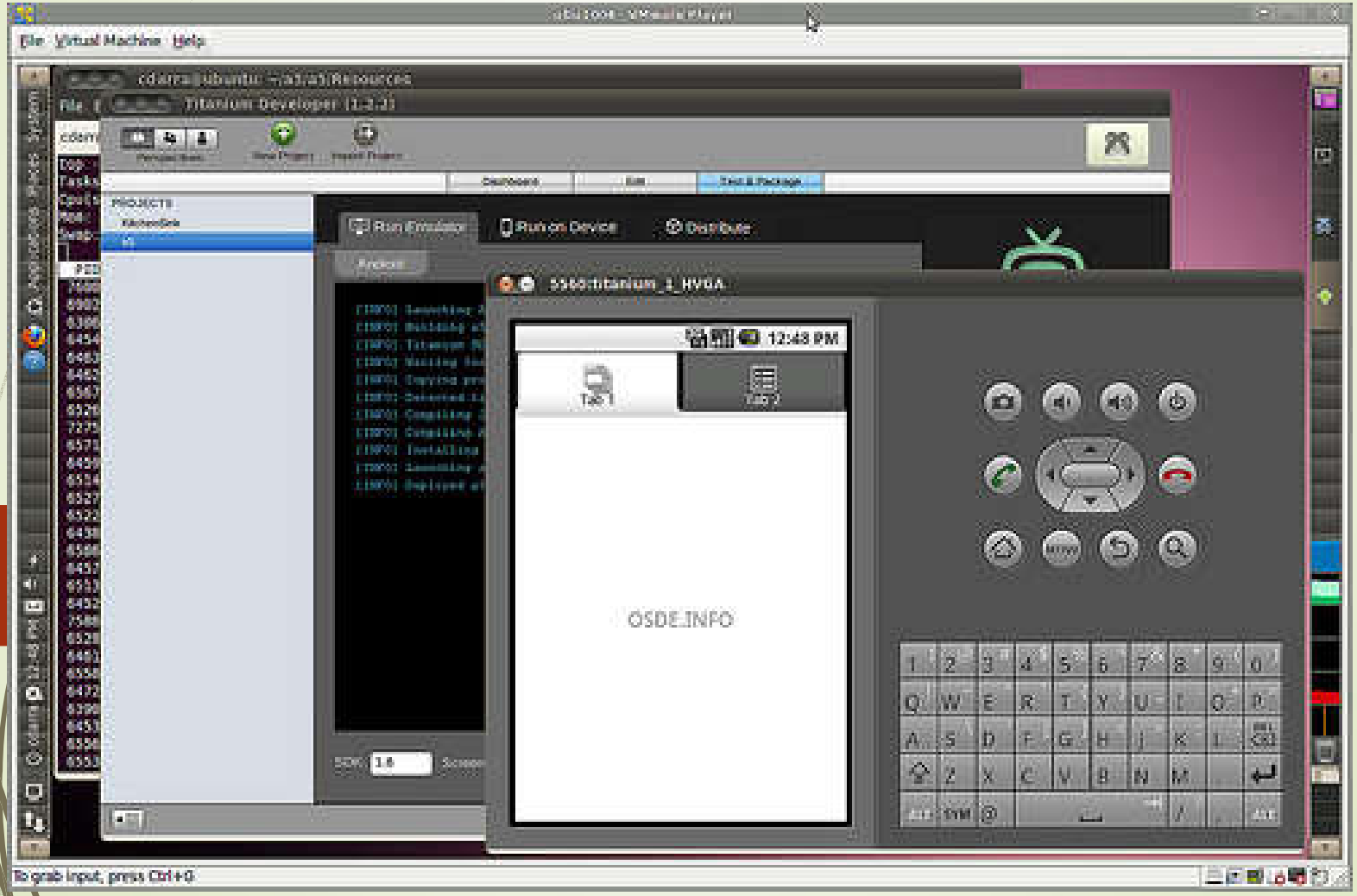
- There is a set of tools that allow the design of mobile applications with ready-made interfaces or web technologies, and then 'translate' the program code so that it is 'understood' by different devices.
- Such an approach allows the rapid creation of applications that are compatible with a large number of devices and offer a rich set of functionality.

Modern technologies for rapid development

Appcelerator Titanium

- Appcelerator Titanium SDK is one of the leading software tools for mobile application development. It has 1.5 million active programmers and there are about 22,000 developed applications .
- Code libraries and supporting platform tools allow developers to focus on creating applications that can be installed on a variety of mobile devices and desktops, and the applications themselves can be coded with languages such as PHP, Python, CCS, HTML, and JavaScript .

Appcelerator Titanium SDK



Modern technologies for rapid development

MIT App Inventor

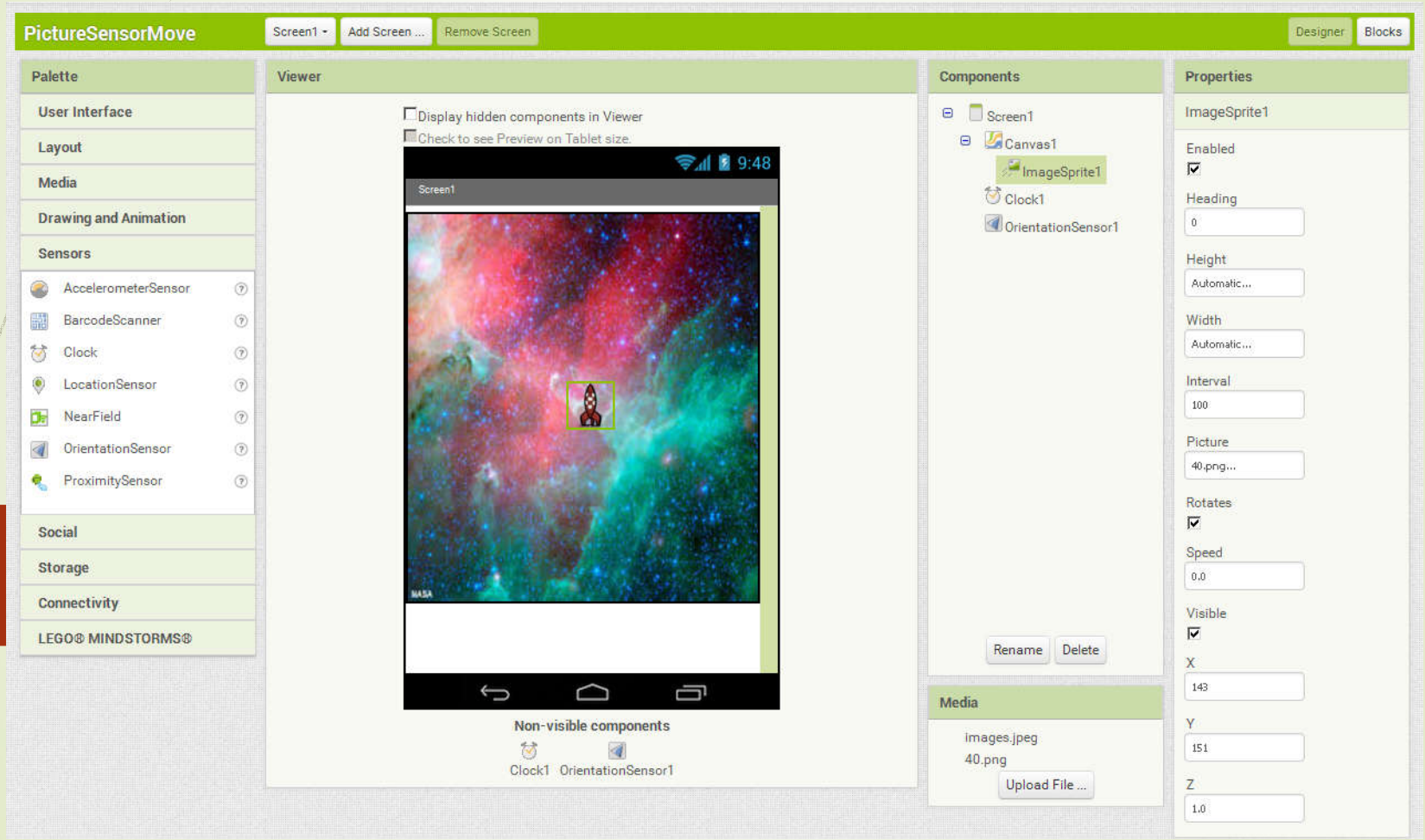
- MIT App Inventor is a cloud-based environment for rapid visual development of applications for the Android operating system .
- App Inventor uses a visual design interface, and the program's operating logic can be created without requiring knowledge of Java programming and the Android SDK .
- Logical blocks connect to each other, similar to arranging a puzzle.
- App Inventor project is created using blocks called components.

Modern technologies for rapid development

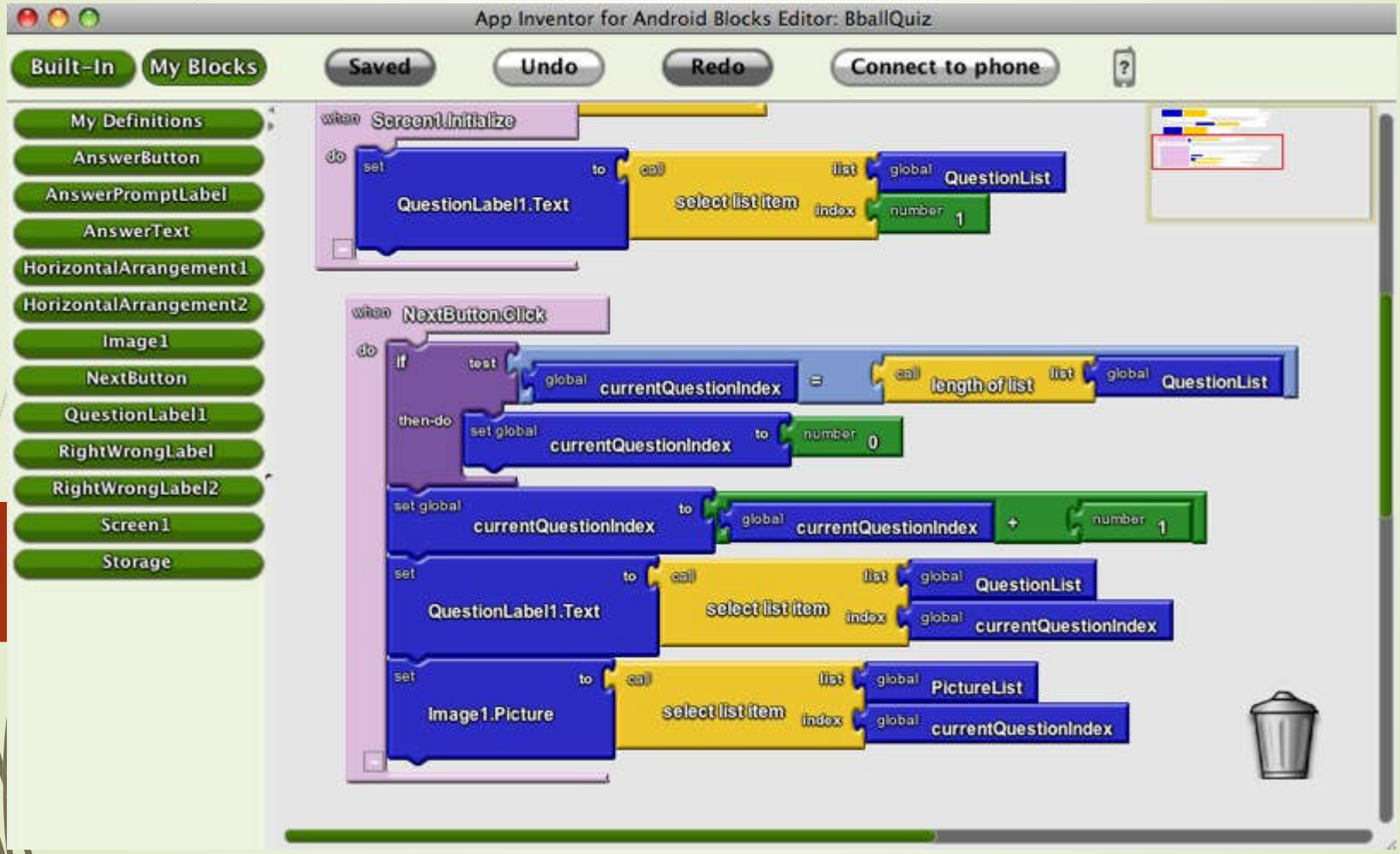
MIT App Inventor

- Such are, for example, Label for displaying text, TextBox for entering information from the user, Camera component, etc.
- In the Pallete box, you can see different categories of components, from basic components such as text input and output, to much more specialized components for media playback and animation, as well as components that serve as an interface to control the device's sensors.
- Components have assigned behaviors, methods, and properties. Some of the properties can be changed and others can be read only.

App Inventor (design window)



App Inventor (Block Editor)



Modern technologies for rapid development

AppMakr

- AppMakr allows the creation of mobile applications without the need for programming.
- For this purpose, the 'point-and-click' method is used, in which it takes ready-made content from a website and embeds it automatically in a desired mobile application.
- Allows users to build Android, iOS and Windows Phone apps. The sfware is free to use.
- Users need no programming experience or knowledge.

AppMakr

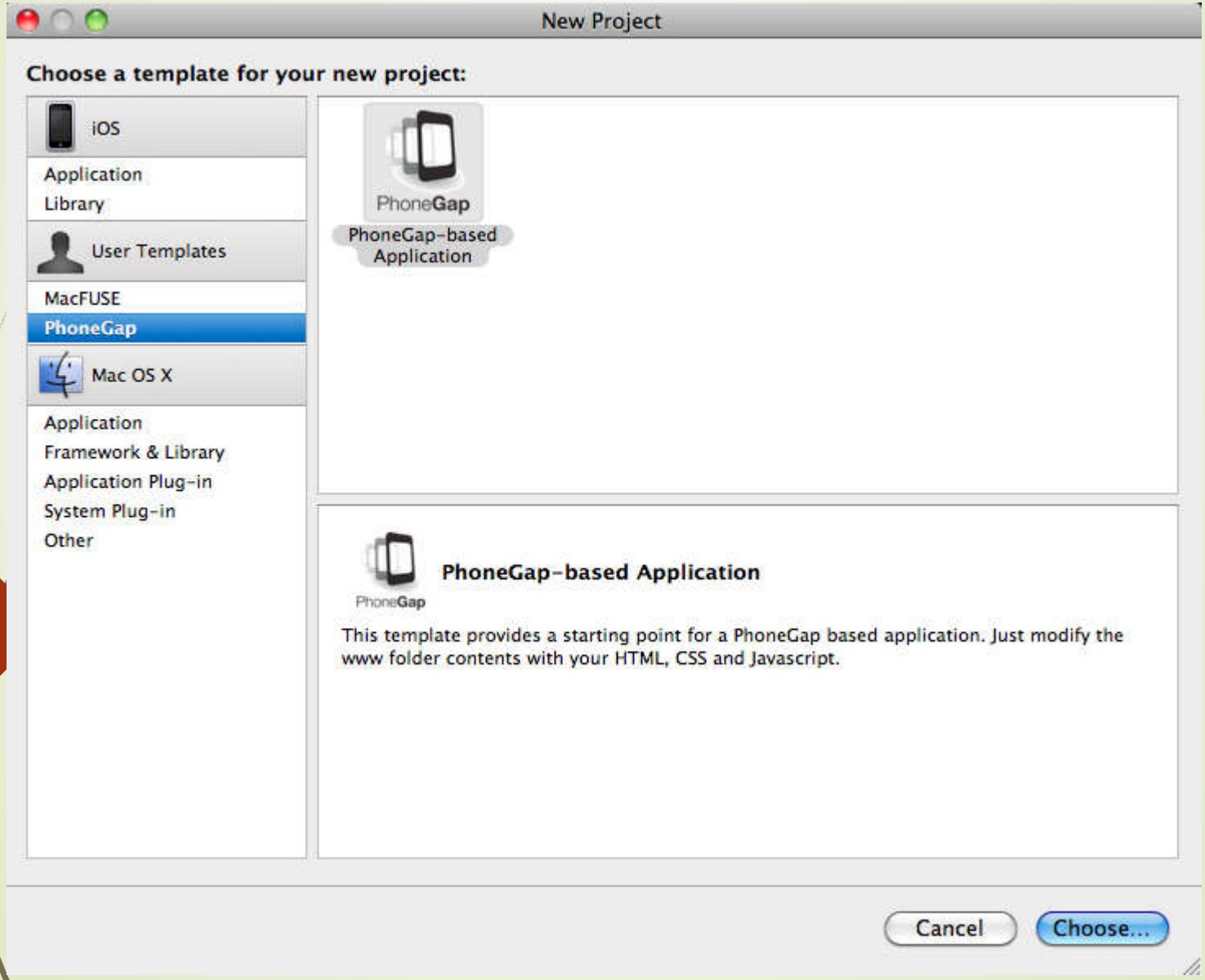

Step 1
Step 2
Step 3
BACK
NEXT
HELP
EXIT
Available in-app functions
Common Functions
Websites
MyBlog
News
Photos
Videos
Contacts
Directions
Calendar
HTML Page
Forms
Docs
LiveChat
Social Feeds
News & Blogs
Photos & Videos
Appearance
Backgrounds
Header
Icons
My Very First App
MyBlog
infinite monkeys
Live Preview
App Saved 32 secs ago
Video Help
OFF ON
MyBlog
Include RSS feed from your favourite blog, or just about any RSS feed. You will need to find the URL (address) for the feed itself, not just the homepage. Look for this symbol or the word 'RSS' or 'feed' on the source site.
Click here to watch the Help Video
Specs / Info
MyBlog
Icon
Background
Feed Url:
RSS feed link
SAVE

Modern technologies for rapid development

PhoneGap

- PhoneGap allows developers to code applications in HTML5 and JavaScript without interfering with basic phone functions.
- The software is open source and provides access to interfaces for various platforms, including iOS, Android, Windows Mobile, Blackberry, and Symbian.
- PhoneGap also enables developers to use built-in device APIs – such as the camera and contact list – directly in JavaScript.
- Ready-made applications are compiled and an installation package is created for each platform.

PhoneGap


New Project
Choose a template for your new project:
iOS
Application
Library
User Templates
MacFUSE
PhoneGap
Mac OS X
Application
Framework & Library
Application Plug-in
System Plug-in
Other
PhoneGap
PhoneGap-based Application
PhoneGap
PhoneGap-based Application
This template provides a starting point for a PhoneGap based application. Just modify the www folder contents with your HTML, CSS and Javascript.
Cancel
Choose...

Modern technologies for rapid development

Sencha Touch

- Sencha offers a different approach to mobile app development.
- Unlike other similar tools, Sencha Touch offers a JavaScript interface for coding web-based applications that do not need approval to be published on the App Store.
- Sencha is a tool that only requires knowledge of JavaScript and HTML.

Modern technologies for rapid development

Sencha Touch

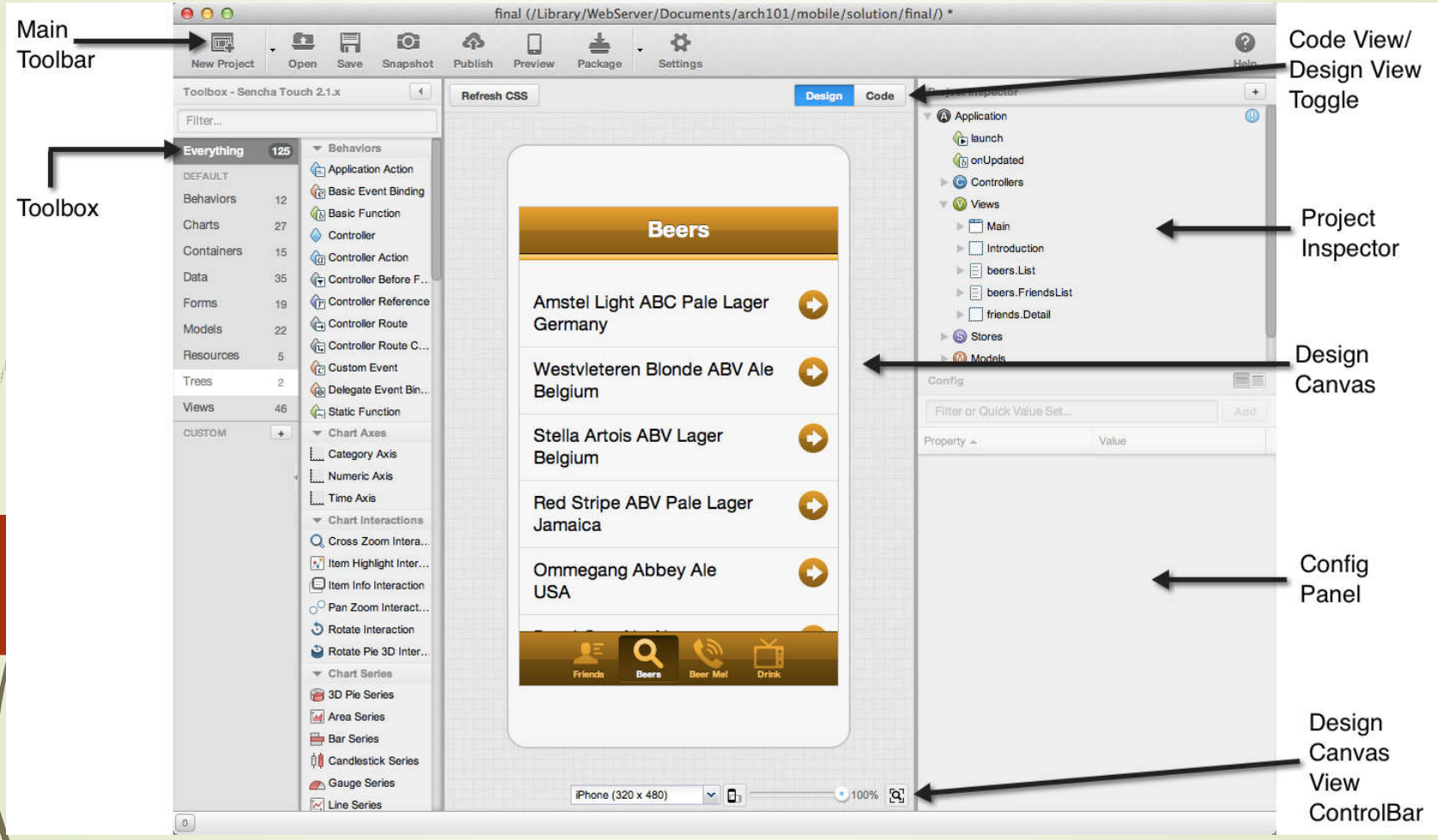
- Sencha Touch enables the development of web-based software for mobile devices that is platform independent, functional as software developed for specific mobile operating systems, and looks and works equally well under its iPhone, Android, Windows Phone operating systems, etc.
- Sencha Touch is the world's first framework built entirely with HTML5, CSS3, and JavaScript, offering powerful controls for user interface construction, flexibility, and optimization.

Modern technologies for rapid development

Sencha Touch

- The specific use of HTML5 provides components such as audio and video, as well as localStorage to save the data offline.
- The extensive use of CSS3 in provides very good styling of all elements of the user interface controls without the need to use images.

Sencha Touch



Modern technologies for rapid development

jQuery Mobile

- jQuery Mobile is an optimized working environment for developing mobile applications for smartphones and tablets.
- It provides a user interface and navigation to run on all popular mobile operating systems. jQuery Mobile is optimized for HTML5 and CSS3.
- The platform comes with the following components: dialog pages, toolbar, buttons, formatted content, forms, sheet views, etc. jQuery Mobile also includes many themes. The development is compatible with all mobile browsers.

Modern technologies for rapid development

jQuery Mobile

- This makes it possible to develop applications with absolutely identical interfaces, regardless of the operating system of the mobile device.
- This is sometimes quite difficult to achieve when developing the use of so-called "thick clients" written respectively in Objective-C, Java or other platform-specific programming languages.

jQuery Mobile