

การพัฒนาแอปพลิเคชัน บนอุปกรณ์เคลื่อนที่แบบข้ามแพลตฟอร์ม



การพัฒนาแอปพลิเคชัน บนอุปกรณ์เคลื่อนที่แบบข้ามแพลตฟอร์ม

เอกรินทร์ วัฒนญลีศสกุล

 **สำนักพิมพ์**
จุฬาลงกรณ์มหาวิทยาลัย
2564

320.-

เอกรินทร์ วัญญูเลิศสกุล

การพัฒนาแอปพลิเคชันบนอุปกรณ์เคลื่อนที่แบบข้ามแพลตฟอร์ม / เอกรินทร์ วัญญูเลิศสกุล

1. โปรแกรมประยุกต์บนอุปกรณ์เคลื่อนที่. 2. การพัฒนาซอฟต์แวร์ข้ามแพลตฟอร์ม.

3. ซอฟต์แวร์ -- การพัฒนา.

005.35

ISBN 978-974-03-4018-8

สปพจ. 2459



สรรคุณค่าวิชาการ สู่สังคม
Knowledge to All
www.cupress.chula.ac.th

ลิขสิทธิ์ของสำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย

การผลิตและการลอกเลียนหนังสือเล่มนี้ไม่ว่ารูปแบบใดทั้งสิ้น

ต้องได้รับอนุญาตเป็นลายลักษณ์อักษรจากสำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย

จัดพิมพ์โดย สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย

พิมพ์ครั้งที่ 1 จำนวน 800 เล่ม พ.ศ. 2564

ผู้จัดจำหน่าย ศูนย์หนังสือจุฬาลงกรณ์มหาวิทยาลัย

ถนนพญาไท เขตปทุมวัน กรุงเทพฯ 10330

สาขา

ศาลาพระเกี้ยว โทร. 0-2218-7000-3 โทรสาร 0-2255-4441

สยามสแควร์ โทร. 0-2218-9881-2 โทรสาร 0-2254-9495

ม.นเรศวร จ.พิษณุโลก โทร. 0-5526-0162-4 โทรสาร 0-5526-0165

ม.เทคโนโลยีสุรนารี จ.นครราชสีมา โทร. 0-4421-6131-4 โทรสาร 0-4421-6135

ม.บูรพา จ.ชลบุรี โทร. 0-3839-4855-9 โทรสาร 0-3839-3239

โรงเรียนนายร้อย จปร. จ.นครนายก โทร. 0-3739-3023 โทรสาร 0-3739-3023

ม.พะเยา จ.พะเยา โทร. 0-5446-6799-800 โทรสาร 0-5446-6798

จัตุรัสจามจุรี (CHAMCHURI SQUARE) ชั้น 4 โทร. 0-2160-5301-2 โทรสาร 0-2160-5304

หัวหมาก โทร. 0-2374-1378 โทรสาร 0-2374-1377

มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน จ.นครราชสีมา โทร. 0-4492-2662-3 โทรสาร 0-4492-2664

Call Center (จัดส่งทั่วประเทศ) โทร. 0-2255-4433 <http://www.chulabook.com>

และเครือข่าย

ร้านค้า หนังสือเข้าชั้นเรียน ติดต่อแผนกขายส่ง สาขาหัวหมาก โทร. 0-2374-1375-6 โทรสาร 0-2374-1375

กองบรรณาธิการ : อรุณี เสมาริ

ออกแบบปกและรูปเล่ม : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย

พิมพ์ที่ : สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย [CUB6311-014] โทร. 0-2218-3562-3 โทรสาร 0-2218-3547

www.cupress.chula.ac.th



คำนำ

Corona SDK เป็นเครื่องมือสำหรับนักพัฒนาแอปพลิเคชันข้ามแพลตฟอร์มที่มีประสิทธิภาพ ใช้ทรัพยากรสำหรับการพัฒนาระบบที่ค่อนข้างน้อยกว่าเครื่องมือส่วนใหญ่ที่มีในท้องตลาดยุคปัจจุบัน เป็นแนวทางในการพัฒนาแอปพลิเคชันที่จะช่วยลดระยะเวลาที่จะต้องใช้ในการพัฒนาแพลตฟอร์ม iOS, Windows และ Android ไปพร้อม ๆ กันได้ด้วยภาษา Lua ซึ่งมีจุดเด่น คือ ง่ายต่อการศึกษา ยืดหยุ่น และทำงานได้รวดเร็ว พร้อมทั้งยังมีข้อมูลข่าวสารอัปเดตตลอดเวลาจากผู้ผลิต และมีชุมชนนักพัฒนาที่เข้มแข็ง ผู้อ่านสามารถศึกษาเรียนรู้ด้วยตนเองได้โดยไม่ยาก ซึ่งแนวทางการนำเสนอในหนังสือเล่มนี้มุ่งเน้นการพัฒนาและเตรียมพื้นฐานในการเป็นนักพัฒนาแอปพลิเคชันที่ดี มีมุมมองในการออกแบบและพัฒนาแอปพลิเคชันอย่างเป็นระบบและถูกต้อง ด้วยการเสริมตัวอย่างการประยุกต์ใช้ และแนวคิดในการออกแบบที่สามารถนำไปใช้งานได้จริง จนกระทั่งสามารถเผยแพร่สู่ผู้ใช้งานผ่านช่องทางดาวน์โหลดของ Google Play Store และ App Store ได้

เอกรินทร์ วัฒนญเลิศสกุล

สารบัญ

	หน้า
บทที่ 1 สถาปัตยกรรมอุปกรณ์เคลื่อนที่	1
วิวัฒนาการการสื่อสารของอุปกรณ์เคลื่อนที่	1
สถาปัตยกรรมอุปกรณ์เคลื่อนที่	5
บทสรุป	14
เอกสารอ้างอิง	16
บทที่ 2 การพัฒนาแอปพลิเคชันข้ามแพลตฟอร์ม	19
แพลตฟอร์ม	19
เฟรมเวิร์ก	20
วงจรชีวิตการพัฒนาสำหรับอุปกรณ์เคลื่อนที่	23
การออกแบบประสบการณ์ผู้ใช้	26
การออกแบบส่วนต่อประสานผู้ใช้	35
บทสรุป	40
เอกสารอ้างอิง	41
บทที่ 3 เครื่องมือสำหรับการพัฒนาแอปพลิเคชัน	43
Corona SDK	43
Corona Editor	50
สภาพแวดล้อมการใช้งาน	57
บทสรุป	63
เอกสารอ้างอิง	65
บทที่ 4 ภาษาที่ใช้ในการพัฒนาแอปพลิเคชัน	67
ภาษา Lua	67
ข้อกำหนด	68
ตัวแปร	69
ชนิดของค่า	71
ตัวดำเนินการ	76
คำสั่งควบคุมทิศทาง	78
บทสรุป	89
เอกสารอ้างอิง	90

สารบัญ (ต่อ)

	หน้า
บทที่ 5 การแสดงผลข้อมูล	91
การแสดงผลข้อมูลพื้นฐาน	91
ข้อความ	95
รูปทรงสี่เหลี่ยม	99
รูปภาพ	102
รูปทรงวงกลม	104
เส้นตรง	105
รูปทรงสี่เหลี่ยมมุมมน	106
กลุ่ม	107
บทสรุป	108
เอกสารอ้างอิง	110
บทที่ 6 การนำเข้าข้อมูล	111
โปรแกรมย่อยและฟังก์ชัน	111
Event Listeners	116
TextField	118
Button	119
SwitchWidget	127
PickerWheel	130
ProgressView	133
TabBar	135
TableView	139
บทสรุป	143
เอกสารอ้างอิง	145
บทที่ 7 ฉาก	147
คอมโพเซอร์	147
การสร้างฉาก	148
การเรียกใช้ฉาก	148
รูปแบบการเปลี่ยนแปลงฉาก	150

สารบัญ (ต่อ)

	หน้า
การสลับฉาก	151
การเรียกใช้ฉากซ้ำ	153
การซ่อนฉาก	157
การลบฉาก	158
การส่งค่าพารามิเตอร์	161
บทสรุป	163
เอกสารอ้างอิง	166
บทที่ 8 ไฟล์และการจัดการฐานข้อมูล	167
ไฟล์	167
เจสัน	174
ฐานข้อมูล	177
สถานะเครือข่าย	182
บทสรุป	185
เอกสารอ้างอิง	187
บทที่ 9 ข้อมูลบริการ MVC	189
MVC	189
ASP.NET MVC	191
คอนโทรลเลอร์	195
วงจรชีวิตของ MVC	198
การกำหนดเส้นทาง	199
แอคชัน	204
วิว	211
โมเดล	215
การส่งคืนค่าแบบเจสัน	218
การร้องขอข้อมูลแบบมีพารามิเตอร์	221
การเพิ่มความมั่นคงปลอดภัยของการร้องขอบริการ	226
บทสรุป	228
เอกสารอ้างอิง	230

สารบัญ (ต่อ)

	หน้า
บทที่ 10 การเชื่อมต่อแอปพลิเคชันกับข้อมูลบริการ	231
การเชื่อมต่อเครือข่าย	231
การออกแบบส่วนต่อประสานผู้ใช้	238
การสร้างฐานข้อมูล	239
การสร้างข้อมูลบริการ	241
การเรียกใช้ข้อมูลบริการ	248
การจัดการข้อมูลตอบรับ	259
บทสรุป	262
เอกสารอ้างอิง	263
บทที่ 11 การประยุกต์ใช้อินเทอร์เน็ตของสรรพสิ่ง	265
อินเทอร์เน็ตของสรรพสิ่ง	265
เอ็มคิวทีที	268
โบรกเกอร์เอ็มคิวทีที	274
ไลบรารีเอ็มคิวทีที	277
การพัฒนาแอปพลิเคชันสำหรับผู้เผยแพร่	278
การพัฒนาแอปพลิเคชันสำหรับผู้ติดตาม	287
บทสรุป	293
เอกสารอ้างอิง	294
บทที่ 12 การเผยแพร่แอปพลิเคชัน	295
การเตรียมข้อมูลสำหรับสร้างชุดโปรแกรมเผยแพร่	295
การเผยแพร่แอปพลิเคชันสำหรับอุปกรณ์ Android	304
การเผยแพร่แอปพลิเคชันสำหรับอุปกรณ์ iOS	315
บทสรุป	329
เอกสารอ้างอิง	332
บรรณานุกรม	333
ดัชนีค้นคำ	337
ประวัติผู้เขียน	343

บทที่

1

สถาปัตยกรรมอุปกรณ์เคลื่อนที่



อุปกรณ์เคลื่อนที่ที่มีใช้งานเริ่มต้นจากในอดีตจนถึงปัจจุบัน มีวิวัฒนาการและแนวคิดในการออกแบบเพื่อตอบสนองสภาพการใช้งานของผู้คนที่วิถีชีวิตที่เปลี่ยนแปลงไปสอดคล้องกับเทคโนโลยีการสื่อสารที่พัฒนาอย่างไม่หยุดนิ่ง ในบทนี้ผู้อ่านจะได้ทราบวิวัฒนาการด้านการสื่อสาร โครงสร้างสถาปัตยกรรมการออกแบบทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ของอุปกรณ์เคลื่อนที่ในแต่ละยุคสมัยเพื่อใช้เป็นแนวทางสำหรับการวิเคราะห์และออกแบบการใช้งานแอปพลิเคชันต่อไป

1.1 วิวัฒนาการการสื่อสารของอุปกรณ์เคลื่อนที่

ในช่วงทศวรรษ ค.ศ. 1970 มาติน คอปเปอร์ (Martin Cooper) วิศวกรของบริษัทโมโตโรลา (Motorola) ซึ่งเป็นบริษัทด้านการสื่อสารข้ามชาติ ซึ่งเป็นที่รู้จักในฐานะผู้ผลิตวิทยุสื่อสาร อุปกรณ์สารกึ่งตัวนำและโทรศัพท์มือถือ ได้คิดค้นโทรศัพท์มือถือรุ่นแรกเพื่อใช้พัฒนาสำหรับการใช้งานในอุปกรณ์เคลื่อนที่ต่าง ๆ โดยได้ต้นแบบที่ใช้ในการทดสอบครั้งแรกในปี ค.ศ. 1974 การประดิษฐ์นี้ถือเป็นจุดเปลี่ยนในการสื่อสารไร้สายซึ่งนำไปสู่วิวัฒนาการของเทคโนโลยีและมาตรฐานมากมาย ด้วยเหตุนี้ระบบการสื่อสารของอุปกรณ์เคลื่อนที่หรือมือถือจึงมีช่วงวิวัฒนาการที่ปรับปรุงเปลี่ยนแปลงหลายช่วงโดยแบ่งออกเป็นรุ่น (Generation) สอดคล้องตามการพัฒนาทางด้านเทคโนโลยีการสื่อสารไร้สายที่เปลี่ยนแปลงไป (Mir M. M. and Kumar S., 2015)

1.1.1 ระบบสื่อสารมือถือยุค 1G

เครือข่ายมือถือยุคแรก (First Generation) หรือยุค 1G ถูกนำไปใช้ในประเทศญี่ปุ่นโดยบริษัท Nippon Telephone และ Telegraph ในโตเกียวระหว่างปี ค.ศ. 1979 ในช่วงต้นทศวรรษ 1980 ได้รับความนิยมในสหรัฐอเมริกา ฟินแลนด์ สหราชอาณาจักร และยุโรป ระบบนี้ใช้สัญญาณแอนะล็อกและมีข้อเสียมากมายเนื่องจากข้อจำกัดทางเทคโนโลยีที่มีในขณะนั้น

1G ใช้เทคโนโลยี AMPS (Advanced Mobile Phone System) เป็นเครือข่ายเสียงเท่านั้น ที่ทำงานบนคลื่นความถี่ 800 MHz AMPS ดำเนินการในลักษณะเดียวกับการส่งสัญญาณวิทยุปกติ ที่ย่านความถี่ 800 MHz ถูกแบ่งออกเป็น 416 ช่องความถี่ประกอบด้วย 395 ช่อง (Channels) สำหรับการใช้งานของเสียง (Voice) และ 21 ช่องสำหรับการควบคุม ผ่าน FDMA โดยแต่ละช่องจะมีความกว้าง 30 KHz อุปกรณ์สื่อสารในยุคนี้มีขนาดใหญ่ และราคาสูง หน้าจอประกอบด้วยตัวเลขเท่านั้น เครือข่ายรองรับจำนวนผู้ใช้งานได้ไม่มากนัก และง่ายต่อการถูกโจรกรรมจากการดักฟัง



ภาพที่ 1.1 โทรศัพท์เคลื่อนที่ในยุค 1G รุ่น Nokia Cityman

ที่มา: Nokiamuseum.info (Online)

1.1.2 ระบบสื่อสารมือถือยุค 2G

ระบบสื่อสารเคลื่อนที่ยุคที่สองได้นำเสนอเทคโนโลยีดิจิทัลใหม่สำหรับการส่งข้อมูลแบบไร้สายหรือที่รู้จักกันในชื่อ Global System for Mobile Communication (GSM) เทคโนโลยี GSM กลายเป็นมาตรฐานพื้นฐานสำหรับการพัฒนาเพิ่มเติมในมาตรฐานอุปกรณ์ไร้สายต่อมาในภายหลัง มาตรฐานนี้สามารถรองรับอัตราการส่งข้อมูลสูงสุด 14.4 ถึง 64kbps ซึ่งเพียงพอสำหรับบริการข้อความ SMS และอีเมล (Doherty J., 2015)

ต่อมาบริษัท Qualcomm ได้เปิดตัวระบบ Code Division Multiple Access (CDMA) ในช่วงปลายปี ค.ศ. 1990 ซึ่งระบบ CDMA มีคุณสมบัติที่ดีกว่า GSM ในด้านการจัดการประสิทธิภาพเชิงสเปกตรัม รองรับจำนวนผู้ใช้และอัตราการส่งข้อมูลที่สูงกว่า ในยุคนี้ยังสามารถแบ่งพัฒนาการด้านเทคโนโลยีออกเป็น 2 ระบบ ประกอบด้วย 2.5G และ 2.75G โดยมีความสามารถในการรองรับอัตราการรับส่งข้อมูลที่สูงขึ้นจนเกิดมีบริการรับส่งข้อมูลการสื่อสารผ่านทางโทรศัพท์มือถือต่าง ๆ เช่น

- GPRS (General Packet Radio Service) สามารถรับส่งข้อมูลได้สูงสุด 171 kbps
- EDGE (Enhanced Data GSM Evolution) พัฒนาขึ้นเพื่อปรับปรุงอัตราการรับส่งข้อมูลสำหรับเครือข่าย GSM EDGE ซึ่งสามารถรองรับอัตราการรับส่งข้อมูลได้มากถึง 473.6 kbps
- CDMA (Code Division Multiple Access) รองรับอัตราการรับส่งข้อมูลที่สูงขึ้น สำหรับเครือข่าย CDMA 2000 มีความสามารถในการรับส่งข้อมูลสูงสุด 384 kbps



ภาพที่ 1.2 อุปกรณ์เคลื่อนที่ในยุค 2G
ที่มา: Flashfly (Online)

1.1.3 ระบบสื่อสารมือถือยุค 3G

ระบบสื่อสารยุคที่สาม เป็นระบบการสื่อสารที่มีความสามารถในการออนไลน์ตลอดเวลา (Always On) อุปกรณ์สื่อสารเคลื่อนที่สามารถเข้าถึงอินเทอร์เน็ตได้ทุกที่ทุกเวลา อัตราการใช้บริการเกิดขึ้นตามปริมาณการใช้งานข้อมูล ผู้ใช้บริการระบบจึงสามารถรับข้อมูลข่าวสารได้ทันทีเมื่อเกิดการสื่อสารในระบบเกิดขึ้น จึงเหมาะกับบริการออนไลน์ต่าง ๆ เช่น อีเมล อินเทอร์เน็ต การสนทนาออนไลน์ บริการวิดีโอสตรีมมิ่ง และโซเชียลมีเดีย ซึ่งแตกต่างกับยุค 2G ที่จะต้องทำการลงทะเบียนเข้าใช้งานเครือข่าย (Log-on) ทุกครั้งเมื่อต้องการใช้งานและระบบจะคิดค่าบริการทันทีที่มีการลงทะเบียนเข้าใช้งาน ซึ่งหากเปรียบเทียบค่าใช้จ่ายและพฤติกรรมความต้องการสื่อสารด้านเครือข่ายสังคมยอดนิยมต่าง ๆ เช่น Hi5, Facebook, LINE และ Youtube จะพบว่าได้ส่งผลให้เกิดกระแสตอบรับและความต้องการใช้งานข้อมูลออนไลน์ที่หลากหลายมากขึ้น รวมถึงเกิดบริการในรูปแบบใหม่ที่เรียกว่า แอปพลิเคชัน



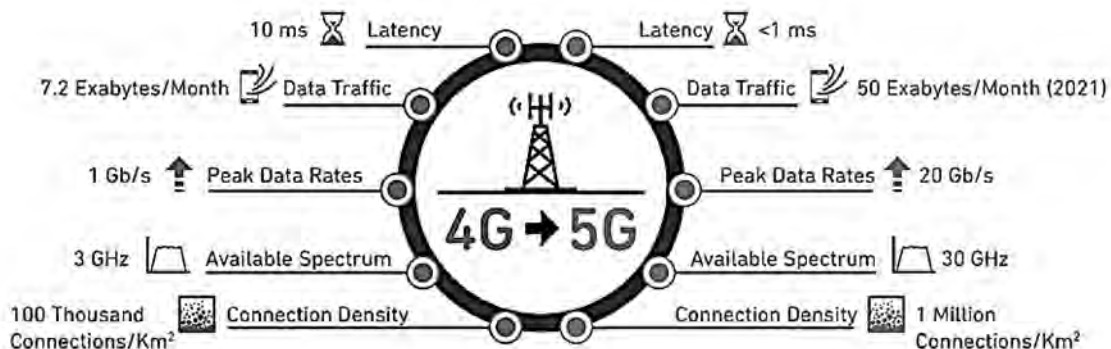
ภาพที่ 1.3 โทรศัพท์เคลื่อนที่ยุค 3G
ที่มา: Ashkan (Online)

1.1.4 ระบบสื่อสารมือถือยุค 4G

ระบบ 4G เป็นรุ่นปรับปรุงระบบเครือข่าย 3G ซึ่งเรียกว่า LTE (Long Term Evolution) ที่พัฒนาโดย IEEE ให้อัตราการส่งข้อมูลที่สูงขึ้นและสามารถรองรับบริการสื่อประสมขั้นสูงได้มากขึ้น ด้วยเทคโนโลยีไร้สายขั้นสูง มีอัตราความเร็วในการรับส่งข้อมูลได้ 20-40 Mbps จึงทำให้สามารถส่งสัญญาณเสียงและข้อมูลพร้อมกันได้อย่างมีประสิทธิภาพ รวมถึงมีบริการเสียงที่สามารถส่งผ่านแพ็คเกจ IP ได้ ทำให้สามารถดูวิดีโอออนไลน์ที่มีความคมชัดสูง การสื่อสารข้ามประเทศแบบเห็นหน้า และการประชุมผ่านโทรศัพท์ (Mobile Teleconferencing) โดยไม่มีค่าใช้จ่าย (Mir M. M. and Kumar S., 2015)

1.1.5 ระบบสื่อสารมือถือยุค 5G

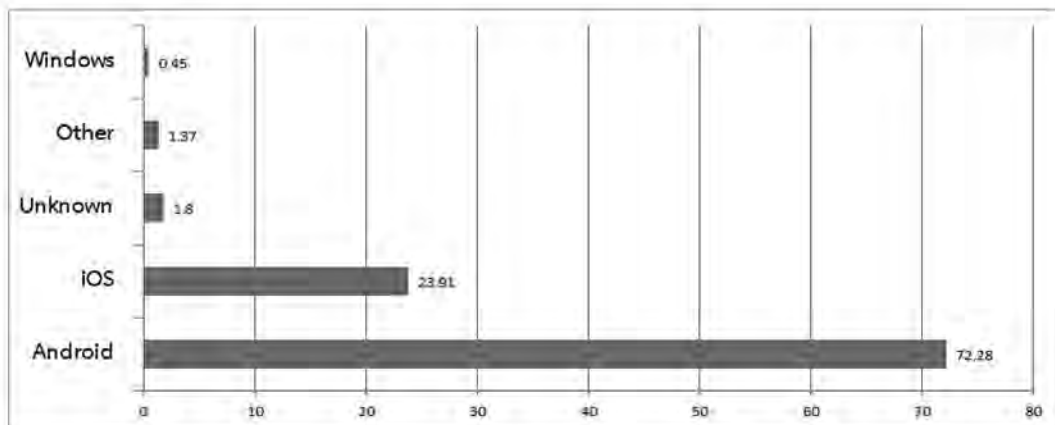
ระบบ 5G จะใช้เทคโนโลยีขั้นสูง ถูกพัฒนาเพื่อใช้สนับสนุนการใช้งานอินเทอร์เน็ตและสื่อประสมที่รวดเร็วสำหรับผู้ใช้ระบบเครือข่าย รองรับการใช้งานอัตราการส่งข้อมูลจำนวนมากเพื่อใช้สำหรับเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่ง (Internet of Things: IoT) รองรับอัตราความเร็วในการสื่อสารสูงสุด 20 Gbps มีค่าความล่าช้า (Latency) ของระยะเวลาการเชื่อมต่อไปยังปลายทางน้อยกว่า 0.001 วินาที ซึ่งสามารถเปรียบเทียบความสามารถของเทคโนโลยี 4G และ 5G ได้ โดยมีรายละเอียดดังภาพที่ 1.4



ภาพที่ 1.4 เปรียบเทียบความสามารถเทคโนโลยี 4G และ 5G
ที่มา: Qorvo (Online)

1.2 สถาปัตยกรรมอุปกรณ์เคลื่อนที่

การศึกษาสถาปัตยกรรมพื้นฐานของอุปกรณ์เคลื่อนที่เป็นสิ่งสำคัญยิ่งสำหรับการใช้เป็นแนวทางในการพัฒนาแอปพลิเคชัน ซึ่งจำเป็นต้องมีความสัมพันธ์และต้องมีการโต้ตอบกับระบบปฏิบัติการด้วย เนื่องจากระบบปฏิบัติการที่แตกต่างกันจะมีวิธีการในการออกแบบและการพัฒนาที่แตกต่างกัน ซึ่งระบบปฏิบัติการที่ใช้ในอุปกรณ์เคลื่อนที่เป็นซอฟต์แวร์ระบบที่ถูกติดตั้งมาพร้อมกับอุปกรณ์เคลื่อนที่ตามลักษณะโครงสร้างของสถาปัตยกรรม โดยจะถูกจัดเก็บไว้ในหน่วยความจำแบบถาวร ระบบปฏิบัติการที่นิยมใช้งานในอุปกรณ์เคลื่อนที่ ได้แก่ Windows Mobile, Pocket PC's OS, Android และ iOS (Hasan M. and Haque A., 2016)



ภาพที่ 1.5 ส่วนแบ่งทางการตลาดของระบบปฏิบัติการของอุปกรณ์เคลื่อนที่ประจำปี ค.ศ. 2018

ที่มา: Statcounter (Online)

1.2.1 สถาปัตยกรรมระบบปฏิบัติการ Android

ระบบปฏิบัติการแอนดรอยด์ (Android) ได้ถือครองส่วนแบ่งการตลาดมากกว่า 70% ของอุปกรณ์เคลื่อนที่แบบสมาร์ทโฟน ทั้งนี้เนื่องจากแอนดรอยด์เป็นระบบปฏิบัติการรหัสเปิดที่มีความยืดหยุ่นสามารถนำไปปรับใช้ในงานเชิงธุรกิจได้หลากหลาย เช่น อุปกรณ์สื่อสารกันทั้งภายในบ้าน เครื่องนำทางติดรถ และอุปกรณ์เครื่องใช้ไฟฟ้าสมัยใหม่ จึงมีพันธมิตรและนักพัฒนามากมาย

แอนดรอยด์ ถูกพัฒนาในปี พ.ศ. 2546 โดย แอนดี้ รูบิน (Andy Rubin) ที่มุ่งมั่นในการที่จะพัฒนาอุปกรณ์เคลื่อนที่ที่มีความชาญฉลาด โดยพัฒนามาจากระบบปฏิบัติการลินุกซ์ และในปี พ.ศ. 2548 บริษัทกูเกิลได้ซื้อเทคโนโลยีแอนดรอยด์มูลค่าหลายพันล้านดอลลาร์เพื่อทำโครงการพัฒนาระบบปฏิบัติการสำหรับอุปกรณ์เคลื่อนที่ โดยในช่วงเวลานั้นยังไม่ค่อยมีคนรู้จักแอนดรอยด์และสิ่งที่ถูกเก็ลกำลังพยายามทำอยู่ จนกระทั่งปี พ.ศ. 2550 กูเกิลได้ประกาศแพลตฟอร์มระบบปฏิบัติการสำหรับอุปกรณ์เคลื่อนที่เป็นครั้งแรก โดยมโนบายส่งเสริมนวัตกรรมบนอุปกรณ์เคลื่อนที่และมอบประสบการณ์การใช้งานที่ดีขึ้นแก่ผู้บริโภคมากกว่าสิ่งที่มีอยู่ในแพลตฟอร์มในยุคสมัยนั้น

สถาปัตยกรรมแอนดรอยด์มีการจัดแบ่งการทำงานออกเป็นชั้น ๆ ที่เรียกว่า เลเยอร์ (Layer) โดยมี เคอร์เนลลินุกซ์ตั้งอยู่ที่เลเยอร์ชั้นล่างสุด โดยด้านบนจะมี Android Runtime และ Native Library ส่วนเลเยอร์ที่สูงในชั้นถัดไปจะประกอบด้วย Application Framework และ Applications ตามลำดับ (Krajci I. and Cummings D., 2014)

1.2.1.1 Applications

เลเยอร์แอปพลิเคชัน (Applications Layer) คือ ชั้นที่อยู่ในระดับสูงสุด เป็นส่วนที่ผู้ใช้งานคุ้นเคยมากที่สุด รองรับความต้องการในการใช้งานอุปกรณ์ทั้งจากแอปพลิเคชันที่แอนดรอยด์รองรับอยู่แล้ว (Native Android Application) และแอปพลิเคชันที่มาจากผู้ผลิตรายอื่น (Third Party Application) เช่น การรับส่งข้อความและอีเมล การท่องอินเทอร์เน็ต แผนที่ และรายชื่อผู้ติดต่อ เป็นต้น แอปพลิเคชันเหล่านี้ส่วนใหญ่เขียนด้วยโปรแกรมภาษา Java โดยทั่วไปแล้วจะอยู่ในไดเรกทอรี data/app

1.2.1.2 Application Frameworks

เลเยอร์แอปพลิเคชันเฟรมเวิร์ก (Application Frameworks) คือ เฟรมเวิร์กของแอนดรอยด์ที่เป็นชุดรวบรวมไลบรารีที่เรียกว่า API ที่ช่วยให้นักพัฒนาสามารถเขียนแอปพลิเคชันได้อย่างรวดเร็วและง่ายดายมากยิ่งขึ้น ประกอบด้วยเครื่องมือสำหรับการออกแบบส่วนต่อประสานผู้ใช้ เช่น ปุ่ม ฟิลด์ข้อความ บานหน้าต่าง รูปภาพและเครื่องมือระบบ การควบคุมโทรศัพท์ และเครื่องเล่นสื่อประสม เป็นต้น ประกอบด้วยส่วนประกอบสำคัญ เช่น

- 1) Activity Manager ทำหน้าที่ให้ข้อมูลเกี่ยวกับและโต้ตอบกับกิจกรรมบริการและกระบวนการที่มี
- 2) Telephony Manager ทำหน้าที่ให้การเข้าถึงบริการโทรศัพท์เช่นเดียวกับข้อมูลสมาชิกบางอย่าง เช่น ข้อมูลผู้ติดต่อและหมายเลขโทรศัพท์
- 3) View System ทำหน้าที่จัดการมุมมองและเค้าโครงที่ประกอบขึ้นเป็นส่วนหนึ่งของส่วนต่อประสานผู้ใช้
- 4) Location Manager ทำหน้าที่ค้นหาที่ตั้งทางภูมิศาสตร์ของอุปกรณ์
- 5) Content Provider คือ ผู้ให้บริการส่วนเนื้อหาที่จะช่วยให้แอปพลิเคชันจัดการการเข้าถึงข้อมูลที่จัดเก็บได้ด้วยตัวเอง และเป็นวิธีการแบ่งปันข้อมูลเพื่อใช้งานร่วมกับแอปพลิเคชันอื่นได้ โดยรวบรวมข้อมูลและกลไกสำหรับการกำหนดความปลอดภัยของข้อมูลไว้ภายใน
- 6) Notification Manager เป็นส่วนที่ควบคุมและแจ้งเตือนให้ผู้ใช้ทราบถึงเหตุการณ์ที่เกิดขึ้นที่แถบสถานะ เช่น การแสดงข้อความหรือสายที่ไม่ได้รับ การเปิดหรือกระพริบไฟ LED บนอุปกรณ์ การสั่น รวมถึงการแจ้งเตือนในลักษณะอื่น ๆ เพื่อแจ้งให้ผู้ใช้ทราบถึงเหตุการณ์ที่เกิดขึ้น
- 7) Resource Manager คือ ส่วนที่ทำหน้าที่ให้การเข้าถึงทรัพยากรในส่วนที่ไม่ใช่โค้ดโปรแกรม เช่น รูปภาพ และไฟล์รูปแบบ (Layout Files)

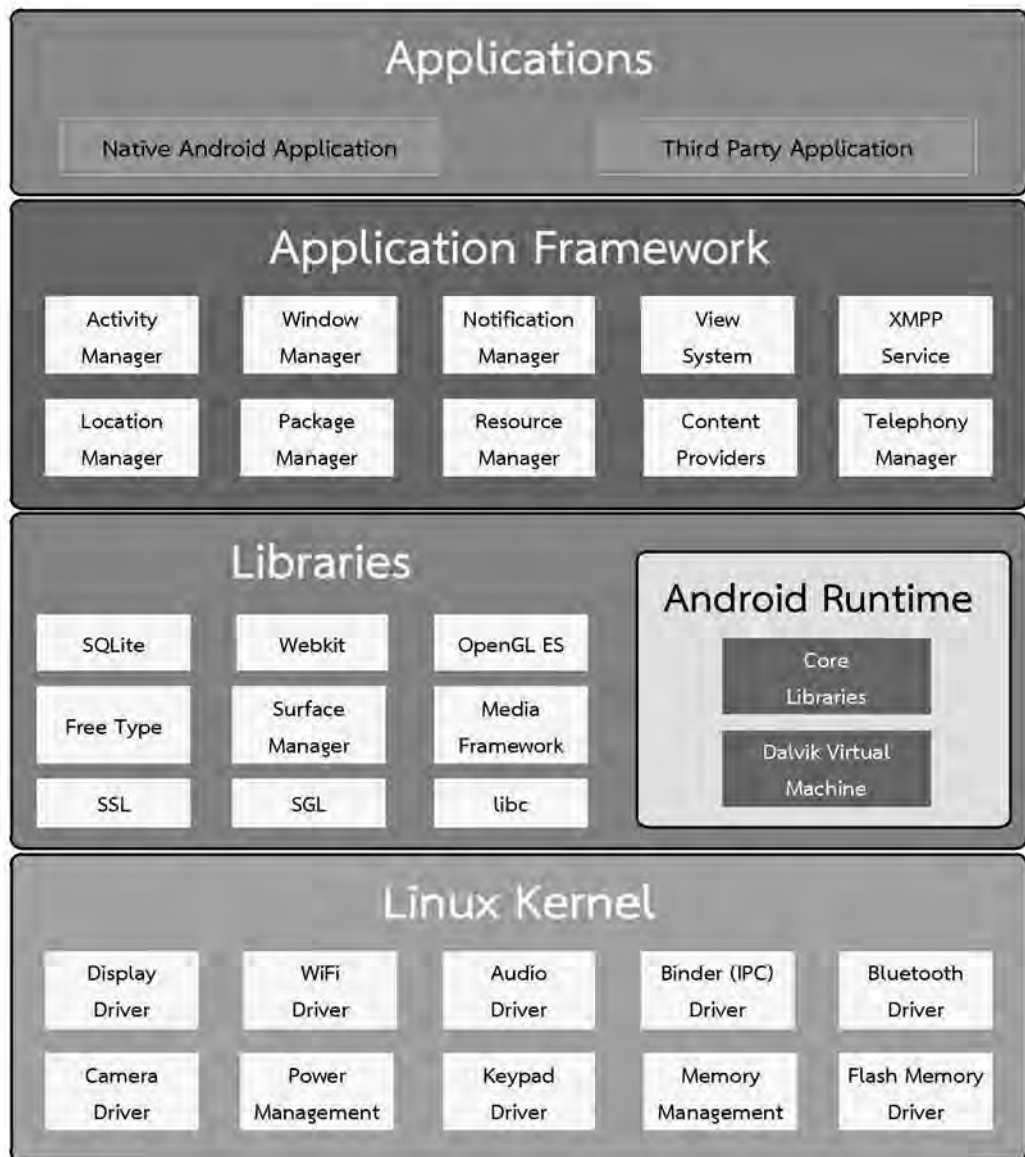
1.2.1.3 Library

เลเยอร์ไลบรารี คือ ส่วนที่ประกอบด้วยชุดของไลบรารีภาษา C และ C++ ที่ถูกนำไปใช้ในเลเยอร์แอปพลิเคชันเฟรมเวิร์กและส่วนประกอบต่าง ๆ ของระบบ โดยได้ทำการปรับให้เหมาะสมสำหรับสภาพฮาร์ดแวร์ของอุปกรณ์ จึงมีความแตกต่างกับชั้นแอปพลิเคชันเฟรมเวิร์กที่เขียนโดยภาษา Java โดยตัวอย่างของไลบรารีที่สำคัญ ได้แก่

- 1) System C Library เป็นกลุ่มของไลบรารีมาตรฐานที่อยู่บนพื้นฐานของภาษา C ไลบรารี
- 2) Open GL (Graphics Library) คือ ไลบรารีที่มีหน้าที่ใช้เพื่อสร้างคอมพิวเตอร์กราฟิกซึ่งแอนดรอยด์มีการสนับสนุนกราฟิก 2D และ 3D ประสิทธิภาพสูง โดยมี OpenGL เป็น API กราฟิกข้ามแพลตฟอร์มที่ออกแบบมาเพื่อใช้งานร่วมกับส่วนอินเตอร์เฟซซอฟต์แวร์มาตรฐานของฮาร์ดแวร์การประมวลผลกราฟิกแบบ 3D
- 3) WebKit คือ ไลบรารีเอ็นจินการเรนเดอร์ที่มีหน้าที่สำหรับสร้างเว็บเพจในมุมมองและหน้าต่าง
- 4) SQLite คือ เอ็นจินฐานข้อมูลเชิงสัมพันธ์แบบรหัสเปิดที่ได้รับการออกแบบมาให้ฝังในอุปกรณ์เคลื่อนที่ เป็นฐานข้อมูลขนาดเล็กที่มีเนื้อที่จำกัดเนื่องจากใช้ทรัพยากรของอุปกรณ์เคลื่อนที่
- 5) FreeType คือ ไลบรารีที่ใช้สำหรับการแสดงผลแบบอักษรแบบเวกเตอร์ (Vector) และบิตแมป (Bitmap)
- 6) Media Frameworks คือ เฟรมเวิร์กที่ใช้สำหรับทำงานร่วมกับไฟล์สื่อประสมจากแหล่งข้อมูลต่าง ๆ เช่น แหล่งข้อมูลภายในแอปพลิเคชันทั้งในรูปแบบไฟล์และสตรีมผ่านการเชื่อมต่อเครือข่ายโดยระบุไว้เป็น URL
- 7) Android Runtime (ART) เป็นสภาพแวดล้อมในการทำงานของแอปพลิเคชันที่ใช้ในระบบปฏิบัติการแอนดรอยด์ โดยมี Dalvik VM ซึ่งเป็นเครื่องเสมือนจริงจำนวนหลาย ๆ เครื่อง ทำงานได้โดยการเรียกใช้ไฟล์ DEX ซึ่งเป็นรูปแบบ Bytecode ที่ออกแบบมาเพื่อบีบอัดคลาส Java ในระบบให้มีขนาดเล็กลงและปรับให้เหมาะสมสำหรับหน่วยความจำของอุปกรณ์เพื่อให้การใช้พลังงานจากแบตเตอรี่อย่างมีประสิทธิภาพสูงสุด

1.2.1.4 Linux Kernel

รากฐานของแพลตฟอร์มแอนดรอยด์ คือ Linux Kernel 2.6 พร้อมการปรับแต่งบางอย่างเพื่อให้เหมาะสมสำหรับการใช้งานในอุปกรณ์เคลื่อนที่ เป็นเลเยอร์ที่อยู่ใกล้กับส่วนที่เป็นฮาร์ดแวร์มากที่สุด จึงมีฟังก์ชันการใช้งานที่เกี่ยวกับการควบคุมและการทำงานของฮาร์ดแวร์ เช่น แป้นพิมพ์ การควบคุมวิทยุ ระบบควบคุมเสียง กล้อง การเชื่อมต่อเครือข่าย และการประมวลผล เป็นต้น

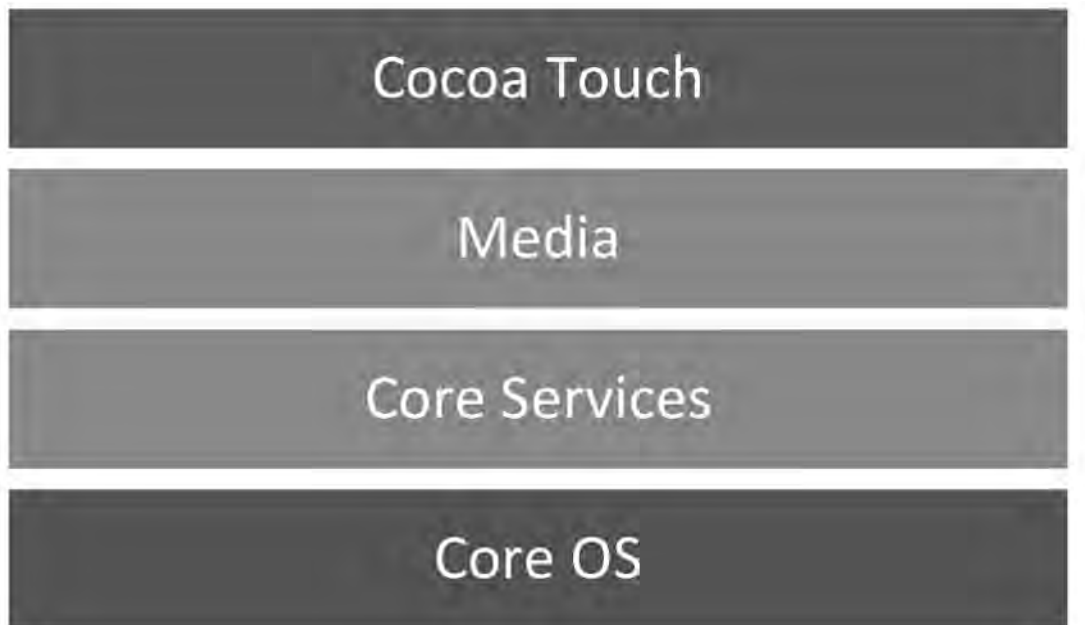


ภาพที่ 1.6 สถาปัตยกรรมระบบปฏิบัติการ Android
ที่มา: ดัดแปลงจาก Burton M. and Felker D., 2015.

1.2.2 สถาปัตยกรรม iOS

iOS คือ ระบบปฏิบัติการที่พัฒนาสำหรับการใช้งานในอุปกรณ์เคลื่อนที่ของบริษัทแอปเปิล (Apple Inc) ซึ่งมีชื่อเดิม คือ ไอโฟนโอเอส (iPhone OS) ซึ่งอุปกรณ์ที่สนับสนุนระบบปฏิบัติการ iOS ทุกเครื่องมีลักษณะทางกายภาพที่เป็นเครื่องคอมพิวเตอร์ จึงมีส่วนที่ประกอบด้วยชุดที่เป็นฮาร์ดแวร์ ได้แก่ โปรเซสเซอร์ หน่วยความจำ อุปกรณ์อินพุต/เอาต์พุต และส่วนจัดเก็บข้อมูล โดยมีรูปแบบการกำหนดค่าทางฮาร์ดแวร์ที่ถูกควบคุมอย่างสมบูรณ์โดยบริษัทแอปเปิล ซึ่งออกแบบมาเพื่อรองรับอุปกรณ์เคลื่อนที่หลักที่มีเพียง 4 รูปแบบ คือ iPhone, iPod Touch, iPad และ iPad mini

อุปกรณ์ iOS ยังมีส่วนประกอบด้านฮาร์ดแวร์ที่ผนวกอยู่ในอุปกรณ์เช่นเดียวกับอุปกรณ์อัจฉริยะอื่น ๆ ได้แก่ กล้อง ระบบเครื่องเสียง ระบบระบุตำแหน่ง ชูวัดความเร่ง และส่วนตรวจวัดสภาพแสง เป็นต้น โดยมีสถาปัตยกรรมการทำงานแบ่งออกเป็น 4 ชั้น ดังรูป (Ramnath R. and Loffing C., 2014)



ภาพที่ 1.7 สถาปัตยกรรมระบบปฏิบัติการ iOS
ที่มา: Ramnath R. and Loffing C., 2014.

1.2.2.1 Cocoa Touch Layer

Cocoa และ Cocoa Touch เป็นเฟรมเวิร์กที่ใช้สำหรับการพัฒนาแอปพลิเคชัน MacOS และ iOS ตามลำดับ ทำหน้าที่ในการจัดการระบบเครือข่ายและส่วนต่อประสานผู้ใช้ ช่วยในการจัดการกับการโต้ตอบแบบจอสัมผัสและเชื่อมต่อระหว่างผู้ใช้งานกับข้อมูลของแอปพลิเคชัน รวมถึงการจัดการกับการทำงานแบบมัลติทาสก์ โดยมีการทำงานร่วมกันกับเฟรมเวิร์ก UIKit ที่ทำหน้าที่ในการจัดเตรียมโครงสร้างพื้นฐานที่จำเป็นสำหรับแอปพลิเคชัน iOS หรือ tvOS โดยมีหน้าต่างและมุมมองสถาปัตยกรรมสำหรับการใช้เป็นส่วนต่อประสานผู้ใช้ที่มีโครงสร้างการจัดการเหตุการณ์สำหรับการนำเสนอแบบหลายจุดสัมผัส (Multi-Touch) และการนำเข้าข้อมูลไปยังแอปพลิเคชันในรูปแบบอื่น ๆ ด้วย จึงทำให้ Cocoa Touch มีความสามารถในการสนับสนุนการเคลื่อนไหว และการสนับสนุนการวาดภาพบนอุปกรณ์เคลื่อนที่ (Dudney B. and Adamson C., 2009)

1.2.2.2 Media Layer

Media Layer เป็นเลเยอร์ที่ทำหน้าที่การแสดงผลภาพกราฟิก ภาพเคลื่อนไหว และเสียง โดย iOS รองรับประเภทสื่อมากกว่า 100 ชนิด ซึ่งครอบคลุมรูปแบบข้อมูลหลายประเภท เช่นเสียง วิดีโอ รูปภาพ และสตรีมมิ่ง

ตารางที่ 1.1 รายการสื่อการสนับสนุนใน iOS

ประเภทสื่อ	รูปแบบสนับสนุน
รูปภาพ	PICT, BMP, GIF, JPEG, TIFF, PNG, DIB, ICO, EPS, PDF
เสียงและข้อมูล	AAC, AIFF, WAVE, uLaw, AC3, MPEG-3, MPEG-4 (.mp4, .m4a), SND, Au, Caf, Adaptive Multi-Rate (.amr)
ภาพเคลื่อนไหว	AVI, AVR, DV, M-JPEG, MPEG-1, MPEG-2, MPEG-4, AAC, 3GPP, OpenDML, 3GPP2, AMC, H.264, iTunes (.m4v), QuickTime (.mov, .qt)
เว็บ/สตรีมมิ่ง	HTTP, RTP, RTSP

1.2.2.3 Core Services Layer

Core Services เป็นเลเยอร์ที่ทำหน้าที่ในการจัดการและบริการระบบพื้นฐานสำหรับใช้ในการพัฒนาแอปพลิเคชัน ประกอบด้วยบริการระบบพื้นฐานที่แอปพลิเคชันทั้งหมดจำเป็นต้องใช้ เช่น การรับรู้ตำแหน่งฐานข้อมูล SQLite สมุดที่อยู่ และสถานะในการใช้งานบริการระบบเครือข่าย เป็นต้น

ตารางที่ 1.2 บริการระบบพื้นฐานในระบบชั้น Core Services Layer

ชื่อบริการ	รายละเอียด
Address Book	ให้บริการการเข้าถึงผู้ติดต่อสมุดรายชื่อของผู้ใช้บนอุปกรณ์ iOS
Networking	ให้บริการของเครือข่ายและกำหนดความพร้อมในการใช้งานระบบเครือข่ายของอุปกรณ์ iOS
File Access	ให้บริการการเข้าถึงการควบคุมและใช้งานของระบบปฏิบัติการในระดับไฟล์
SQLite	การให้บริการฐานข้อมูลขนาดเล็ก ซึ่งรองรับการใช้งานฐานข้อมูลที่อยู่ภายในแอปพลิเคชัน
Core Location	ใช้สำหรับกำหนดตำแหน่งและทิศทางของอุปกรณ์ iOS
Net Services	การให้บริการการเชื่อมต่อ Wi-Fi หรืออุปกรณ์เคลื่อนที่
Core Data	จัดการโมเดลการออกแบบ Model-View-Controller
Core Foundation	จัดการและกำหนดประเภทข้อมูลในรูปแบบของสตริง ตัวเลข วันที่ ข้อมูลคอลเลกชัน และอื่น ๆ ตามมาตรฐานสากลด้วยเทคโนโลยีบันไดและสตริงแบบ Unicode รวมถึงการทำงานของพอร์ตเรด กระบวนการการทำงานแบบวนซ้ำ
Quick Look	การแสดงตัวอย่างของไฟล์แบบเต็มหน้าจออย่างรวดเร็ว ช่วยให้ผู้ใช้สามารถหมุนและครอบตัดรูปภาพ ทำเครื่องหมายบนไฟล์ PDF หรือตัดต่อวิดีโอและคลิปเสียงได้ทันทีโดยไม่ต้องเปิดแอปพลิเคชัน
Social	การสนับสนุนการใช้บริการเครือข่ายสังคม การดึงข้อมูลสำหรับการรวมบัญชีเครือข่ายสังคมของผู้ใช้เข้ากับแอปพลิเคชันที่พัฒนา
Security	บริการด้านการรักษาความปลอดภัยของระบบในอุปกรณ์เคลื่อนที่
WebKit	บริการการแสดงผลข้อมูล การสร้าง และการแก้ไขเนื้อหาในเอกสาร HTML

1.2.2.4 Core OS

Core OS เป็นเลเยอร์ที่อยู่ชั้นล่างสุดของสถาปัตยกรรม iOS ทำหน้าที่รับผิดชอบในการจัดการหน่วยความจำ การจัดสรรทรัพยากร และการคืนเนื้อที่หน่วยความจำเมื่อแอปพลิเคชันทำงานเสร็จสิ้น นอกจากนี้ยังเป็นส่วนที่ทำหน้าที่บริหารจัดการไฟล์และไฟล์ระบบของอุปกรณ์เคลื่อนที่ การจัดการเครือข่าย และงานระบบปฏิบัติการอื่น ๆ โดย Core OS สามารถสื่อสารโต้ตอบโดยตรงกับฮาร์ดแวร์ได้ ประกอบด้วยส่วนประกอบหรือคอมโพเนนต์ต่าง ๆ เช่น OS X Kernel, Mach 3.0, Berkeley Standard Distribution และ Keychain เป็นต้น

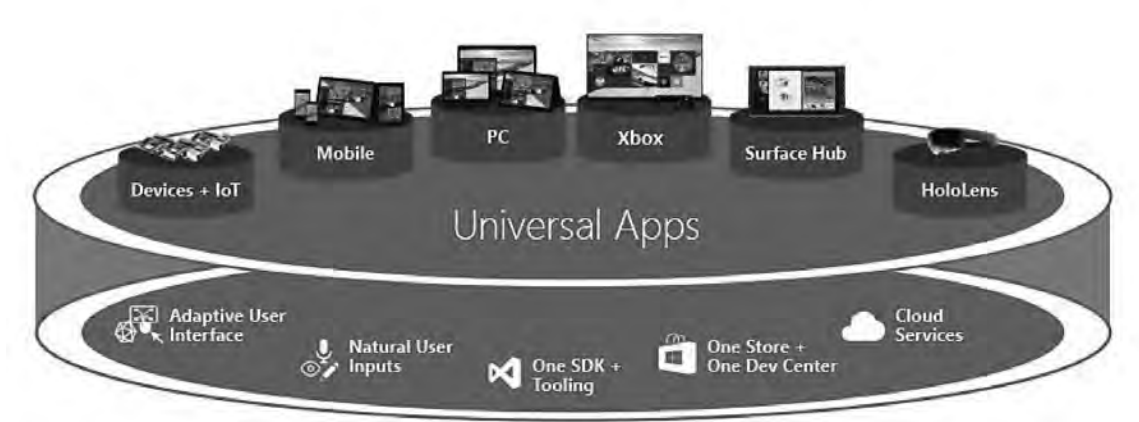
ตารางที่ 1.3 คอมโพเนนต์ของเลเยอร์ Core OS

คอมโพเนนต์	รายละเอียด
OS X Kernel	ทำหน้าที่รับผิดชอบในทุกด้านของระบบปฏิบัติการ
Mach 3.0	เซตย่อยของเคอร์เนล OS X ที่รับผิดชอบในการเรียกใช้แอปพลิเคชันภายในกระบวนการที่แยกต่างหาก
Berkeley Standard Distribution	ภายใต้สภาพแวดล้อมของเคอร์เนลภายในระบบปฏิบัติการ MacOS ส่วนของ BSD จะเป็นผู้รับผิดชอบต่อการทำงานไดรเวอร์และอินเตอร์เฟซระดับต่ำของระบบปฏิบัติการ UNIX
Sockets	เป็นส่วนหนึ่งของ CFNetwork Framework ที่ใช้สำหรับการเข้าถึงบริการเครือข่าย BSD, การร้องขอเพื่อใช้บริการในโพรโทคอล HTTP และ FTP
Security	Security Framework จัดเตรียมฟังก์ชันสำหรับการทำหน้าที่เข้ารหัสและถอดรหัสข้อมูล รวมถึงการสื่อสารและโต้ตอบกับพวงกุญแจ (Keychain) ของ iPhone เพื่อเพิ่ม ลบ และแก้ไขรายการ
Power Management	คุณสมบัติในการจัดการประหยัดพลังงานโดยการปิดคุณสมบัติฮาร์ดแวร์ใด ๆ ที่ไม่ได้ใช้งานในปัจจุบัน
Keychain	ระบบพวงกุญแจที่ช่วยเก็บข้อมูลรหัสบัตรเครดิตและรหัสผ่าน รวมถึงการสร้างรหัสผ่าน และการเชื่อมต่อด้านความปลอดภัยกับอุปกรณ์ iOS และเป็นส่วนหนึ่งของ Security Framework สำหรับการจัดการและรักษาความปลอดภัยข้อมูล
File System	เป็น System Framework ที่ช่วยให้นักพัฒนาสามารถเข้าถึงส่วนการทำงานต่าง ๆ ภายใต้สภาพแวดล้อมของเครื่องมือที่ใช้ในการพัฒนาระบบปฏิบัติการ UNIX

1.2.3 สถาปัตยกรรมระบบปฏิบัติการ Windows Mobile

ไมโครซอฟท์ (Microsoft) ได้เริ่มพัฒนาระบบปฏิบัติการสำหรับอุปกรณ์เคลื่อนที่ในช่วงต้นปี พ.ศ. 2547 เพื่อหวังใช้เพื่อแข่งขันกับเทคโนโลยี Palm PC ซึ่งกำลังเป็นที่นิยมในขณะนั้นในนามของ Windows CE ต่อมาได้พัฒนาปรับเปลี่ยนและปรับปรุงเป็นระบบปฏิบัติการใหม่ในชื่อ Windows Mobile และสามารถวางจำหน่ายได้ในปี พ.ศ. 2551

Windows Phone ถูกวางจำหน่ายครั้งแรกในเดือนตุลาคมปี พ.ศ. 2553 โดยมีเวอร์ชันเริ่มต้นที่ Windows Phone 7 ซึ่งเป็นเวอร์ชันที่นับต่อมาจาก Windows Mobile 6.5 และได้มีการพัฒนาอย่างต่อเนื่องเป็นระบบปฏิบัติการ Windows 8 Mobile และ Windows 10 Mobile ตามลำดับ ซึ่งตั้งแต่ในระบบปฏิบัติการ Windows 8 ไมโครซอฟท์ได้เพิ่มความสามารถในการใช้งานแอปพลิเคชันเดียวที่สามารถรองรับอุปกรณ์ที่หลากหลาย เช่น คอมพิวเตอร์ส่วนบุคคล โทรศัพท์ แท็บเล็ต และอุปกรณ์สื่อประสมอื่น ๆ โดยเรียกเทคโนโลยีที่ใช้ในการพัฒนาแอปพลิเคชันในรูปแบบนี้ว่า Universal Windows Platform (UWP)



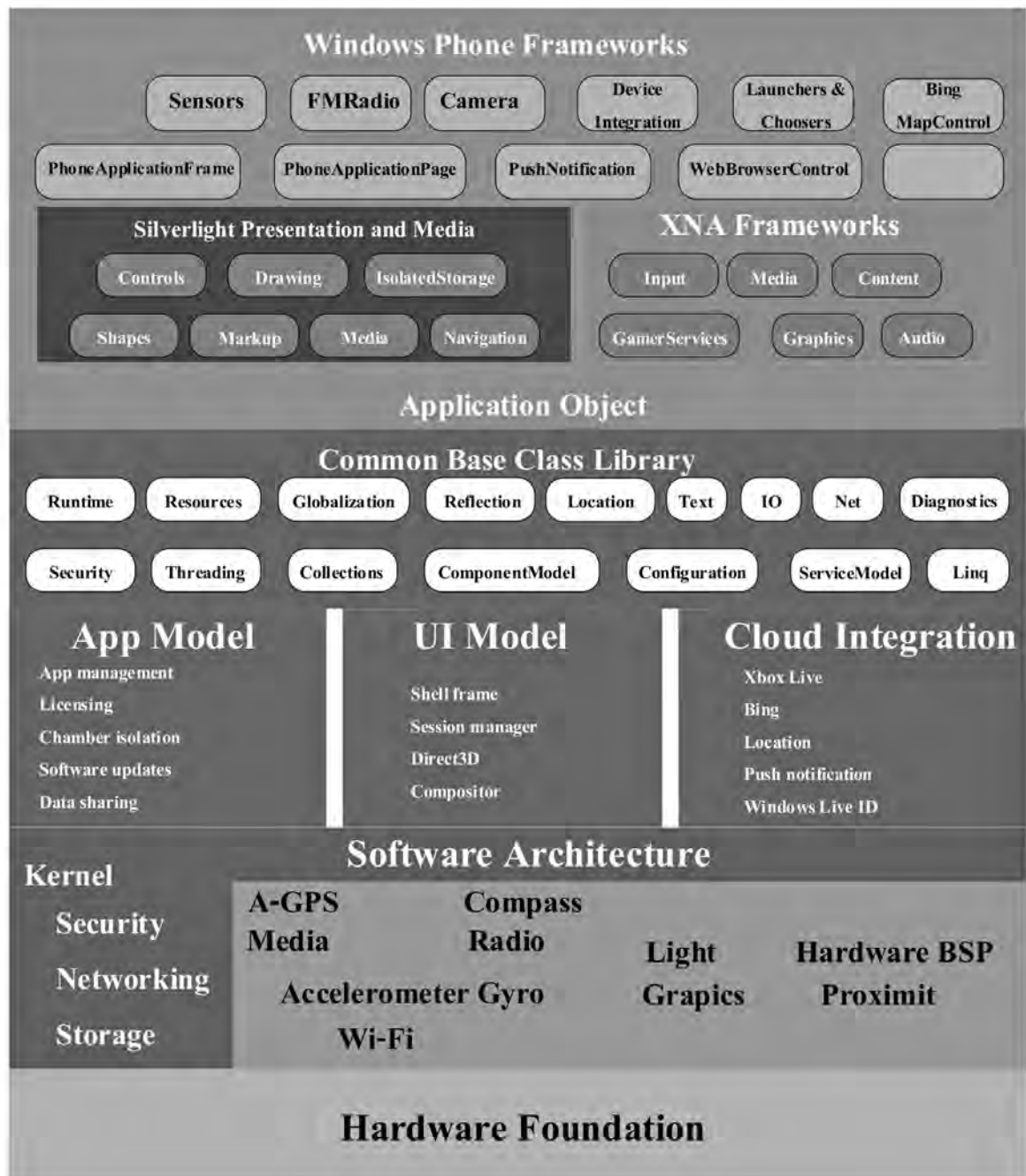
ภาพที่ 1.8 Universal Windows Platform

ที่มา: Ekuam M., 2016.

Windows Phone เป็นระบบปฏิบัติการที่ถูกออกแบบสำหรับใช้งานกับอุปกรณ์เคลื่อนที่แบบหน้าจอสัมผัส แอปพลิเคชันถูกพัฒนาด้วยโค้ดเทคโนโลยี Microsoft .NET Framework เช่น ภาษา C#, Visual Basic และ J# (Gronli T. M., et al., 2014)

ส่วนต่อประสานผู้ใช้ในระบบปฏิบัติการ Windows Phone ใช้รูปแบบที่เรียกว่า Metro UI ทำให้มีความแตกต่างจากระบบปฏิบัติการอื่น ๆ ที่มีอยู่ในท้องตลาด เพื่อสร้างประสบการณ์ใช้งานในรูปแบบที่แตกต่างและอัตลักษณ์ในการใช้งาน โดยมีหลักการที่สำคัญที่สุดในการออกแบบ คือ การทำให้เนื้อหาและข้อมูลโดดเด่นและชัดเจนขึ้นได้ด้วยการกำจัดสายเส้น สี แสง และเงา ที่มาประดับตกแต่งในแอปพลิเคชัน ซึ่งได้รับแรงบันดาลใจมาจากป้ายต่าง ๆ ในที่สาธารณะ เช่น ป้ายจราจร ป้ายประชาสัมพันธ์ในสนามบิน และสถานีรถไฟ เป็นต้น มีวัตถุประสงค์เพื่อให้ผู้คนที่เห็นสามารถเข้าใจได้อย่างรวดเร็วในระยะเวลาอันสั้น

สำหรับแนวคิดในการออกแบบสถาปัตยกรรมในระบบปฏิบัติการ Windows Phone ได้มีการจัดแบ่งส่วนการทำงานออกเป็น 4 ส่วนสำคัญ ประกอบด้วย Task Host, Core Application, Platform Service และ Base OS Service



ภาพที่ 1.9 สถาปัตยกรรมระบบปฏิบัติการ Windows Phone
ที่มา: Redda Y. A., 2012.

หลังจากที่ Windows 10 Phone ได้ถูกเผยแพร่ออกมาแล้วเจอความไม่ลงตัว และทำให้ไมโครซอฟท์ประกาศยุติการพัฒนาระบบปฏิบัติการไปแล้ว โดยจะสนับสนุนการใช้งานต่อไปอีกสักกระยะหนึ่ง โดยรุ่นสุดท้ายที่อัปเดต คือ Windows 10 Mobile, Version 1709 โดยออกมาตั้งแต่ 10 กรกฎาคม พ.ศ. 2561 และกำหนดวันสิ้นสุดการสนับสนุนในเดือนธันวาคม พ.ศ. 2561 (Steve Litchfield, 2017)

1.2.4 ระบบปฏิบัติการ BlackBerry

BlackBerry OS ได้รับการพัฒนาโดย Research in Motion (RIM) สำหรับอุปกรณ์เคลื่อนที่สมาร์ทโฟน BlackBerry และอุปกรณ์แท็บเล็ต โดยระบบปฏิบัติการ BlackBerry OS รุ่น 1.0 เปิดตัวในเดือนมกราคม พ.ศ. 2542 ซึ่งเป็นส่วนหนึ่งของอุปกรณ์วิทยุติดตามตัว (Pager) และอีเมลของ BlackBerry จึงนับได้ว่าหนึ่งในจุดแข็งหลักของอุปกรณ์ BlackBerry คือ ความสามารถในการจัดการอีเมลขององค์กร เหมาะสำหรับผู้ใช้งานที่ให้ความสำคัญกับการใช้งานบริการรับส่งอีเมลบ่อย ๆ โดยไม่ต้องคอยกดปุ่มตรวจสอบ BlackBerry จะใช้รหัสประจำตัวของอุปกรณ์ที่เรียกว่า Pin Code ซึ่งเป็นตัวเลขผสมกับตัวอักษร ดังนั้น BlackBerry แต่ละเครื่องจึงถูกกำหนดหมายเลขประจำตัวส่วนบุคคล (Personal Identification Number: PIN) ที่ไม่ซ้ำกันเพื่อทำให้สามารถระบุตัวตนของอุปกรณ์ BlackBerry แต่ละเครื่องได้ และทำให้มั่นใจได้ว่าอีเมลหรือข้อความที่ถูกส่งไปยังบุคคลนั้นถูกส่งไปอย่างถูกต้อง (Okediran O. O., et al., 2014)

ระบบปฏิบัติการ BlackBerry ยังสนับสนุน Java Mobile Information Device Profile (MIDP) และ Wireless Application Profile (WAP) โดยสามารถใช้โพรโทคอลเหล่านี้ในการซิงโครไนซ์ผ่าน BlackBerry Enterprise Server ได้ โดยในระหว่างการซิงโครไนซ์ตัวเปลี่ยนเส้นทางจะเชื่อมโยง PIN กับบัญชีอีเมล

บทสรุป

สถาปัตยกรรมอุปกรณ์เคลื่อนที่มีวิวัฒนาการตามความเจริญก้าวหน้าของเทคโนโลยีที่เป็นอยู่ตามช่วงขณะนั้น โดยเรียกชื่อตามรุ่นหรือยุคที่ใช้ในการพัฒนา เริ่มตั้งแต่ยุค 1G จนถึงยุคปัจจุบันที่กำลังเข้าสู่ยุค 5G ซึ่งรองรับเทคโนโลยีการสื่อสารไร้สายอย่างสมบูรณ์แบบและมีความเร็วสูงที่สุดมากถึง 20 Gbps แต่หากพิจารณาถึงสถาปัตยกรรมทางด้านฮาร์ดแวร์ คือ การมุ่งเน้นที่ตัวอุปกรณ์ที่ใช้งานจะพบว่า ยังสามารถแบ่งได้ตามระบบปฏิบัติการที่ทำงานสัมพันธ์กับอุปกรณ์เคลื่อนที่โดยตรง จึงสามารถแบ่งได้ 4 รูปแบบตามการพัฒนาอุปกรณ์เคลื่อนที่ที่เป็นที่นิยมใช้งาน ได้แก่ สถาปัตยกรรม Android, สถาปัตยกรรม iOS, สถาปัตยกรรม Windows Mobile และสถาปัตยกรรม BlackBerry

ความเสื่อมถอยในการความนิยมใช้งานเริ่มแสดงให้เห็นได้อย่างชัดเจนจากจำนวนของผู้ใช้งาน BlackBerry ที่ถูกกลุ่มผู้ผลิตอุปกรณ์เคลื่อนที่แนวคิดใหม่ ๆ ที่เน้นความสะดวกสบาย รูปแบบการใช้ชีวิตของคนยุคใหม่ การวางแผนการตลาด และการสนับสนุนนักพัฒนาแอปพลิเคชัน ซึ่งเป็นเหตุผลหลักที่ทำให้ผู้ใช้งานเริ่มเปลี่ยนใจใช้งาน iOS และ Android มากขึ้นอย่างเห็นได้ชัดเจน BlackBerry ต้องปิดตัวลง และถึงแม้ Windows Mobile จะมีความแข็งแกร่งในการเป็นผู้เชี่ยวชาญด้านการออกแบบระบบปฏิบัติการที่มีความสามารถเข้าถึงผู้ใช้งาน

คอมพิวเตอร์ส่วนบุคคลส่วนใหญ่ได้แต่รูปแบบการใช้งานอุปกรณ์เคลื่อนที่นั่นมีรูปแบบเฉพาะตัว และมีวัตถุประสงค์แตกต่างกันโดยสิ้นเชิง ประกอบกับระดับความนิยมใช้งาน iOS และ Android มีความเติบโตสูงขึ้นอย่างรวดเร็ว ดังนั้น ระบบปฏิบัติการ Windows Mobile จึงได้ยุติการพัฒนาในรุ่น Windows Mobile 10 กระแสการพัฒนาแอปพลิเคชันในปัจจุบันจึงมุ่งพัฒนาไปที่ระบบปฏิบัติการ 2 ผู้ผลิตเท่านั้น คือ Android และ iOS

เอกสารอ้างอิง

- Ashkan. **Blackberry vs iPhone**. (Online) February 23, 2011. (cited December 31, 2018). Available from: <https://blogs.ubc.ca/ashkanj/2011/02/23/blackberry-vs-iphone>.
- Burton M. and Felker D. (2015). **Android App Development for Dummies**. 2th ed. John Wiley & Sons.
- Doherty J. (2015). **Wireless and Mobile Device Security**. United State: Jones & Bartlett.
- Dudney B. and Adamson C. (2009). **iPhone SDK development**. **Pragmatic Bookshelf**.
- Ekuam M. (2016). **What's a Universal Windows Platform (UWP) App**. Microsoft Developer Network.
- Flashfly. **รีวิว Nokia 3310 (2017) จากเว็บต่างประเทศ พร้อมเปรียบเทียบกับรุ่นต้นแบบจากปี 2000**. (Online) March 1, 2017. (cited December 31, 2018). Available from: <http://www.flashfly.net/wp/175708>.
- Gronli T. M., et al. (2014, May). **Mobile Application Platform Heterogeneity: Android vs Windows Phone vs iOS vs Firefox OS**. In 2014 IEEE 28th International Conference on Advanced Information Networking and Applications (pp. 635-641). IEEE.
- Hasan M. and Haque A. **Mobile Application Development Approaches : Recommendation for E-commerce Enterprises**. Vantaa: Laurea University of Applied Sciences, 2016.
- Krajci I. and Cummings D. (2014). **Android on X86: An Introduction to Optimizing for Intel Architecture**. Apress.
- Mir M. M. and Kumar S. (2015). **Evolution of Mobile Wireless Technology from OG to 5G**. International Journal of Computer Science and Information Technologies, 6(3): 2545-2551.
- Nokiamuseum.info. **Nokia Cityman**. (Online) May 1, 1987. (cited December 31, 2018). Available from: <http://nokiamuseum.info/nokia-cityman>.
- Okediran O. O., et al. (2014). **Mobile Operating Systems and Application Development Platforms: A Survey**. International Journal of Advanced Networking and Applications, 6(1): 2195.
- Qorvo. **Getting to 5G: Comparing 4G and 5G System Requirements**. (Online) September 7, 2017. (cited December 31, 2018). Available from: <https://www.qorvo.com/design-hub/blog/getting-to-5g-comparing-4g-and-5g-system-requirements>.
- Ramnath R. and Loffing C. (2014). **Beginning IOS Programming for Dummies**. New Jersey: John Wiley & Sons.
- Redda Y. A. (2012). **Cross Platform Mobile Applications Development: Mobile Apps Mobility**. Trondheim: Norwegian University of Science and Technology.
- Statcounter. **Mobile Operating System Market Share Thailand**. (Online) n.d. (cited December 31, 2018). Available from: <http://gs.statcounter.com/os-market-share/mobile/thailand>.

Steve Litchfield. **Windows 10: the differences between Mobile and Desktop.** (Online) January 10, 2017. (cited December 31, 2018). Available from: http://allaboutwindowsphone.com/features/item/21946_Windows_10_whats_common_betwee.php.

บทที่

2

การพัฒนาแอปพลิเคชัน ข้ามแพลตฟอร์ม

ในบทนี้จะเป็นการแนะนำการพัฒนาแอปพลิเคชันข้ามแพลตฟอร์ม ความหมายของแพลตฟอร์มและเฟรมเวิร์ก แนวคิดและการออกแบบ ประสบการณ์ของผู้ใช้งาน รวมถึงการออกแบบส่วนต่อประสานกราฟิกผู้ใช้ เพื่อให้แอปพลิเคชันที่พัฒนาตรงตามความต้องการตามที่ได้ออกแบบไว้ และสร้างความประทับใจในการใช้งานมากที่สุด

2.1 แพลตฟอร์ม

แพลตฟอร์ม (Platform) ในความหมายตามพจนานุกรม คือ เวที แท่น ยกพื้น พื้นยกระดับ พื้นเวที ชานชาลา (บุญเสริม ลัมมนาวงศ์, 2557)

แพลตฟอร์ม หมายถึง การออกแบบแนวคิด แนวคิด ความคิด รูปแบบหรือโมเดล (Baldwin C. Y. and Woodard C. J., 2009)

คำจำกัดความของ “แพลตฟอร์ม” ที่ใช้ในงานด้านเทคโนโลยีคอมพิวเตอร์จะใช้เพื่อแบ่งแยกรูปแบบการใช้งานระบบคอมพิวเตอร์ เช่น คอมพิวเตอร์ส่วนบุคคลจะประกอบด้วยแพลตฟอร์ม ฮาร์ดแวร์ (เครื่องคอมพิวเตอร์) และซอฟต์แวร์ (ระบบปฏิบัติการ) ซึ่งในการพัฒนาแอปพลิเคชันในปัจจุบันจะพิจารณาจากระบบปฏิบัติการที่เรียกว่า แพลตฟอร์มเดสก์ท็อป (Desktop Platform) ประกอบด้วยระบบปฏิบัติการที่นิยมใช้งานใน

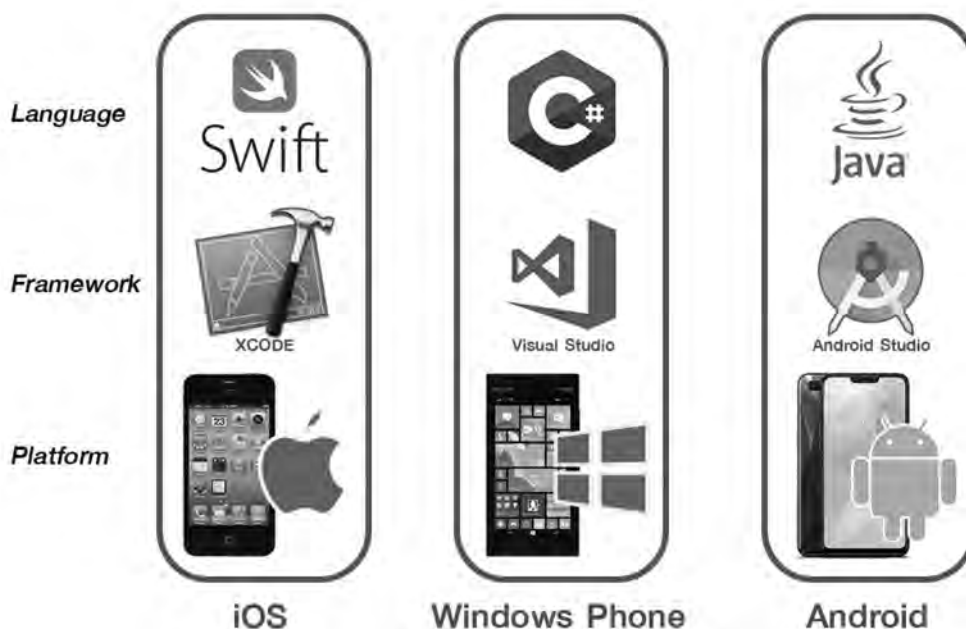
อันดับต้น ๆ คือ Windows, Linux และ MacOS ปัจจุบันแพลตฟอร์มยังมีขอบเขตความหมายที่ครอบคลุมถึงระบบปฏิบัติการของอุปกรณ์เคลื่อนที่ด้วย ดังนั้น หากจะกล่าวถึงแพลตฟอร์มของอุปกรณ์เคลื่อนที่ (Mobile Platform) ซึ่งเป็นที่รู้จักและใช้งานกันแพร่หลายจึงหมายถึงสถาปัตยกรรมของอุปกรณ์เคลื่อนที่ เช่น iOS, Android, Windows Phone, BlackBerry, และ Symbian (Krajci I. and Cummings D., 2014)

2.2 เฟรมเวิร์ก

สำหรับการพัฒนาแอปพลิเคชันของอุปกรณ์เคลื่อนที่จะมีรูปแบบเทคโนโลยีหรือเครื่องมือที่ใช้ในการสร้างหรือบำรุงรักษาแอปพลิเคชันจะเรียกว่า “เฟรมเวิร์ก (Framework)” โดยเฟรมเวิร์กแต่ละแบบจะมีความสามารถตอบสนองการใช้งานต่อแพลตฟอร์มที่แตกต่างกันไป ซึ่งสามารถแบ่งตามวิธีการที่ใช้ในการพัฒนาแอปพลิเคชันสำหรับอุปกรณ์เคลื่อนที่ได้ 3 แบบหลัก คือ แบบเนทีฟ แบบข้ามแพลตฟอร์ม และแบบผสม ส่วนการพัฒนาแอปพลิเคชันด้วยเว็บแอปพลิเคชันจะไม่ขอกล่าวรวมในหัวข้อนี้เนื่องจากไม่รองรับการใช้งานเครื่องมือของอุปกรณ์เคลื่อนที่อย่างแท้จริง

2.2.1 การพัฒนาแอปพลิเคชันอุปกรณ์เคลื่อนที่แบบเนทีฟ

การพัฒนาแอปพลิเคชันสำหรับอุปกรณ์เคลื่อนที่แบบเนทีฟ (Native Mobile Application Development) คือ การพัฒนาแอปพลิเคชันโดยใช้เครื่องมือที่ออกแบบมาสำหรับพัฒนาอุปกรณ์ในแต่ละแพลตฟอร์มโดยเฉพาะ เช่น Android ใช้ภาษา Java และโปรแกรม Android Studio ในการพัฒนา ส่วน iOS ใช้ภาษา Objective-C หรือ Swift และใช้โปรแกรม XCode เป็นเครื่องมือสำหรับการพัฒนาแอปพลิเคชัน เป็นต้น



ภาพที่ 2.1 เฟรมเวิร์กสำหรับการพัฒนาแอปพลิเคชันอุปกรณ์เคลื่อนที่แบบเนทีฟ

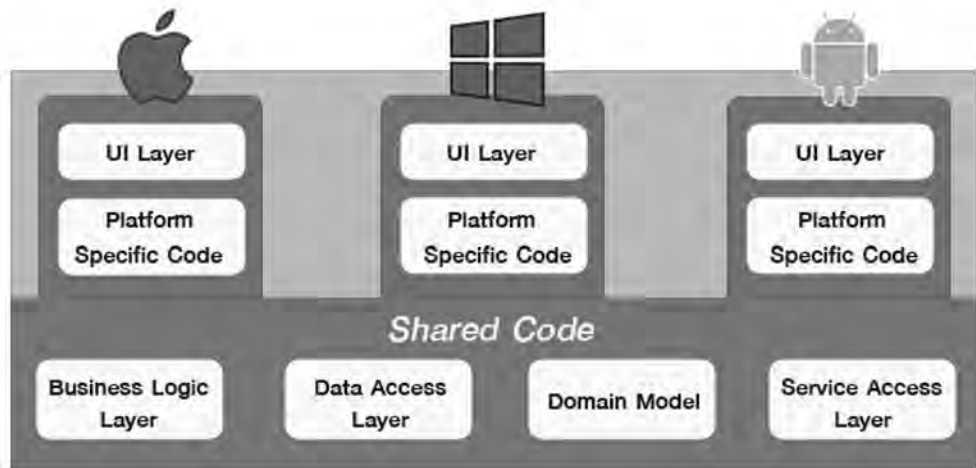
คุณสมบัติหลักทั่วไปของแอปพลิเคชันแบบเนทีฟ คือ การเรียกใช้ความสามารถและเข้าถึงฮาร์ดแวร์ของอุปกรณ์ได้อย่างไม่จำกัด สนับสนุนส่วนติดต่อผู้ใช้และการโต้ตอบทั้งหมดที่มีอยู่ในสภาพแวดล้อมการทำงานของอุปกรณ์เคลื่อนที่ได้อย่างครอบคลุม (Jobe W., 2013)

2.2.2 การพัฒนาแอปพลิเคชันอุปกรณ์เคลื่อนที่แบบข้ามแพลตฟอร์ม

การพัฒนาแอปพลิเคชันอุปกรณ์เคลื่อนที่แบบข้ามแพลตฟอร์ม (Cross Platform Mobile Application Development) เป็นรูปแบบในการพัฒนาแอปพลิเคชันโดยใช้การเขียนโค้ดเพียงครั้งเดียวแต่สามารถใช้งานได้ในทุกแพลตฟอร์ม จึงสามารถรองรับการใช้งานระบบปฏิบัติการต่าง ๆ เช่น Android, iOS และ Windows Phone ได้ การพัฒนาแอปพลิเคชันเคลื่อนที่แบบข้ามแพลตฟอร์มจะช่วยแก้ปัญหาเรื่องระยะเวลาที่ต้องใช้ในการพัฒนาแอปพลิเคชันในรูปแบบเนทีฟ ซึ่งมีการใช้โปรแกรมภาษาและเครื่องมือที่แตกต่างกัน ดังนั้น ระยะเวลาที่ใช้ในการพัฒนาแอปพลิเคชันในแต่ละแพลตฟอร์มจึงแปรผันโดยตรงกับจำนวนของแพลตฟอร์มที่ต้องการพัฒนา (Khandozhenko R., 2014)

การพัฒนาแอปพลิเคชันแบบข้ามแพลตฟอร์มจึงเป็นที่นิยมและแพร่หลายในงานด้านธุรกิจเป็นอย่างมากเพราะสามารถลดต้นทุนในกระบวนการผลิตและช่วยส่งเสริมต่อการดำเนินกิจการทางธุรกิจ สามารถช่วยลดระยะเวลาในการพัฒนาและลดค่าใช้จ่ายในการจ้างนักพัฒนางานได้ ส่งผลให้อัตราส่วนผลตอบแทนจากการลงทุน (Return on Investment: ROI) มีค่าสูงมากขึ้น ซึ่งเครื่องมือที่นิยมใช้ในการพัฒนาแอปพลิเคชันอุปกรณ์เคลื่อนที่แบบข้ามแพลตฟอร์ม ได้แก่ Corona SDK, Xamarin, Cocos Studio และ Flutter เป็นต้น

ความสามารถในการเขียนโค้ดโปรแกรมเพียงตัวเดียวให้สามารถใช้งานแอปพลิเคชันได้หลายแพลตฟอร์มเป็นคุณสมบัติมาจากสถาปัตยกรรมการแชร์โค้ดข้ามแพลตฟอร์ม (Cross-Platform Code Sharing Architecture) โดยทำการแยกส่วนของโค้ดที่ใช้ในการพัฒนาออกเป็นชั้น (Layer) ประกอบด้วย ชั้นตรรกะทางธุรกิจ (Business Logic Layer) ชั้นการเข้าถึงข้อมูล (Data Access Layer) ชั้นการเข้าถึงบริการ (Service Access Layer) และแบบจำลองโดเมน (Domain Model) ซึ่งในการพัฒนาแอปพลิเคชันมักนิยมจะใช้เทคนิค MVC (Model-View-Controller) ช่วยแยกชั้นในการพัฒนาแต่ละชั้นออกจากกัน (รายละเอียด MVC จะกล่าวในบทที่ 9) หลังจากนั้นจะเป็นหน้าที่ของเฟรมเวิร์กที่ใช้ในการพัฒนาแอปพลิเคชันข้ามแพลตฟอร์มจะเลือกใช้ API และไลบรารีต่าง ๆ ที่ได้จากระบบปฏิบัติการของแต่ละแพลตฟอร์ม เพื่อให้ได้โค้ดเฉพาะของแพลตฟอร์มนั้น ๆ (Platform Specific Code) รวมทั้งได้ส่วนต่อประสานผู้ใช้งานที่จะนำไปใช้ในอุปกรณ์เคลื่อนที่ของแต่ละแพลตฟอร์ม (Petzold C., 2016)



ภาพที่ 2.2 สถาปัตยกรรมการแชร์โค้ดข้ามแพลตฟอร์ม

2.2.3 การพัฒนาแอปพลิเคชันอุปกรณ์เคลื่อนที่แบบผสม

ในยุคแรกของการพัฒนาแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ได้มีแนวคิดในการประยุกต์ใช้เว็บแอปพลิเคชันเพื่อใช้งานบนอุปกรณ์โดยเรียกใช้งานผ่านเว็บเบราว์เซอร์ แม้วิธีการนี้จะมีข้อดีของการเขียนโค้ดเพียงครั้งเดียวคล้ายกับแบบข้ามแพลตฟอร์ม แต่รูปแบบการแสดงผลและความน่าใช้ได้ถูกจำกัดด้วยแถบและเครื่องมือต่าง ๆ ของเว็บเบราว์เซอร์ ปัจจุบันจึงได้มีการพัฒนาเทคโนโลยีการพัฒนาเว็บไซต์ด้วย HTML5 ซึ่งเป็นภาษาที่ถูกพัฒนาต่อมาจากภาษา HTML โดยมีการปรับปรุงเพิ่มความสามารถและคุณสมบัติให้แสดงผลได้ดีในทุกเว็บเบราว์เซอร์รองรับการแสดงผลของเนื้อหา เสียง รูปภาพ และภาพเคลื่อนไหวโดยไม่ต้องใช้โปรแกรมประเภท Adobe Flash ซึ่งแพลตฟอร์ม iOS ไม่สนับสนุนการแสดงผลแล้ว

การพัฒนาแอปพลิเคชันสำหรับอุปกรณ์เคลื่อนที่แบบผสม (Hybrid Mobile Application Development) จึงเป็นการผสมผสานความสามารถดั้งเดิมของอุปกรณ์เคลื่อนที่เข้ากับความสามารถในการพัฒนาแอปพลิเคชันโดยใช้เทคโนโลยีเว็บด้วยภาษา HTML5 เพื่อให้สามารถสร้างแอปพลิเคชันที่มีเนื้อหาเดียวกันโดยเขียนโค้ดในครั้งเดียวแต่สามารถแสดงผลได้ในหลายแพลตฟอร์ม โดยใช้ WebView, CSS และ JavaScript เป็นตัวช่วยในการพัฒนาและสนับสนุนการแสดงผล แอปพลิเคชันจึงมีความสวยงามและน่าใช้ อย่างไรก็ตาม การพัฒนาแอปพลิเคชันอุปกรณ์เคลื่อนที่แบบผสมยังมีข้อด้อยในด้านการควบคุมและการใช้งานอุปกรณ์ (Redda Y. A., 2012) และการใช้งานทรัพยากรที่ถูกใช้มากกว่าเมื่อเทียบกับการพัฒนาแบบเนทีฟ (Griffith C., 2017)



ภาพที่ 2.3 เปรียบเทียบการพัฒนาแอปพลิเคชันอุปกรณ์เคลื่อนที่

ตัวอย่างเฟรมเวิร์กที่ใช้สำหรับการพัฒนาแอปพลิเคชันบนอุปกรณ์เคลื่อนที่แบบผสม ได้แก่ Appceleator Titanium, Cordova (PhoneGap) และ Ionic เป็นต้น

2.3 วงจรชีวิตการพัฒนาสำหรับอุปกรณ์เคลื่อนที่

วงจรชีวิตการพัฒนาระบบ (System Development Life Cycle: SDLC) สำหรับอุปกรณ์เคลื่อนที่มีกระบวนการดำเนินการที่ไม่แตกต่างจาก SDLC สำหรับการพัฒนาแอปพลิเคชันบนเว็บ หรือเดสก์ท็อป กล่าวคือ ประกอบด้วยหลักการสำคัญ 5 ขั้นตอน ดังนี้ (Burns A., Umbaugh B. and Dunn C., 2016)