บทความนี้อธิบายเกี่ยวกับสถาปัตยกรรมของ software-intensive system โดยซอฟต์แวร์ที่ถูกมองจากหลายมุมมองก็จะมีข้อกังวลจากหลาย stakeholders และเพื่อรับมือกับความต้องการที่หลากหลาย ในแต่ละมุมมองที่เราจะพูดถึงต่อไปนี้จะเป็นตัวแทนของมัน

การใช้ diagram เพื่อแสดงโครงสร้างของระบบบ่อยครั้งทำให้ผู้เขียนลำบากที่แสดงความซับซ้อนมากขึ้นในแผนภาพเดียว ซึ่งโครงสร้างของซอฟต์แวร์อาจถูกทำให้เสียหายจากการออกแบบระบบที่ได้ไปได้ไกลเกินไป เพื่อแก้ไขปัญหานี้ จึงใช้การจัดรายละเอียดของโครงสร้างซอฟต์แวร์โดยใช้มุมมองหลาย ๆ ประการ แต่ละตัวจะเน้นที่ชุดหนึ่งของความกังวลที่เฉพาะเจาะจง

การออกแบบซอฟต์แวร์เกี่ยวข้องกับการนำเสนอแนวคิด, การแยกออกและรวมเข้าด้วยกัน, และสไตล์และลักษณะทางศิลปะ ในการอธิบายซอฟต์แวร์ที่อยู่ในโครงสร้าง ใช้โมเดลที่ประกอบด้วยมุมมองหลาย ๆ ประการ หรือมุมมอง เพื่อจะสามารถเผชิญหน้ากับโครงสร้างที่ใหญ่และท้าทาย โดยโมเดลนี้ประกอบด้วย 5 มุมมองหลัก คือ Logical view ,Process view ,Physical view, Development view และ Scenario

- The Logical Architecture โครงสร้างทางตรรกะมีบทบาทสำคัญในการสนับสนุนความต้องการเกี่ยวกับการทำงาน
- The Process Architecture โครงสร้างกระบวนการให้ความสนใจถึงความต้องการที่ไม่ใช่ฟังก์ชัน, เช่น ประสิทธิภาพและความพร้อมใช้งาน
- The Development Architecture โครงสร้างการพัฒนาเน้นที่การจัดระเบียบโมดูลซอฟต์แวร์จริงในสภาพแวดล้อมการพัฒนาซอฟต์แวร์ โดยซอฟต์แวร์ถูกแบ่งแพคเกจเป็นชิ้นเล็ก ๆ
- The Physical Architecture โครงสร้างทางกายภาพให้ความสนใจเป็นหลักความต้องการที่ไม่ใช่ฟังก์ชันของระบบ เช่น ความพร้อมใช้งาน, ความน่าเชื่อถือ (ทนต่อข้อผิดพลาด), ประสิทธิภาพ
- Scenarios ในโครงสร้างนี้, มุมมองต่าง ๆ ถูกออกแบบให้ทำงานร่วมกันได้ผ่านการใช้สถานการณ์ที่สำคัญ ซึ่งเป็นตัวอย่างของการใช้งานทั่วไป คือการใช้สถานการณ์ทำให้เราเข้าใจว่าแต่ละมุมมองมีหน้าที่ที่ชัดเจนและทำงานร่วมกันได้ดี เพื่อให้โครงสร้างซอฟต์แวร์มีประสิทธิภาพและตอบสนองต่อความต้องการของระบบ

มุมมองต่าง ๆ ไม่ได้เป็นอิสระหรือแยกจากกันทั้งหมด สิ่งที่ปรากฏในมุมมองหนึ่งมีการเชื่อมโยงกับสิ่งที่ปรากฏในมุมมองอื่น ๆ ตามกฎการออกแบบและเครื่องหมายทางการออกแบบ

ไม่ทุกระบบซอฟต์แวร์ต้องใช้ทุกมุมมองใน 4+1 ถ้ามีมุมมองที่ไม่ได้ให้ประโยชน์ เช่น physical view จะถูกตัดออกหากมีเพียงหนึ่งโปรเซสเซสเซอร์ และ process view ถูกตัดออกหากมีโปรแกรมเดียว สำหรับระบบขนาดเล็กมาก มีโอกาสที่ logical view และ development view จะมีความคล้ายคลึงกันมาก จนไม่จำเป็นต้องแยกกัน แต่ scenario มีประโยชน์ในทุกรูปแบบของสถาปัตยกรรม

สรุปคือ 4+1 คือโมเดลที่ใช้ได้ดีในโครงการขนาดใหญ่ โดยที่แต่ละฝ่ายสามารถเข้าใจโครงสร้างซอฟต์แวร์ตามความต้องการของตนได้ มุมมองต่าง ๆ ช่วยให้ทุกคนเข้าใจได้ง่ายขึ้น วิศวกรระบบมองจากทางกายภาพและกระบวนการ ลูกค้าและผู้ใช้งานมองจากมุมมองตรรกะ ผู้จัดการโครงการและพนักงานกำหนดค่าซอฟต์แวร์มองจากมุมมองการพัฒนา