

# Chapter 57 - Separating Game Code From Platform Code

---

A large port becomes manageable when the game code stops knowing how the machine underneath it is built.

In this case study the shared game code calls a small set of platform contracts. The IE side supplies the contracts by talking to the real machine: keyboard FIFO, File I/O, Voodoo, audio voices, monotonic time, and save storage.

## 57.1 The Contract Line

---

The platform line is not a trick to hide the hardware. It is a way to keep the hardware in one place.

The game core asks for services such as:

Need	IE implementation idea
Time	Read the monotonic microsecond MMIO pair and convert to nanoseconds
Input	Drain the keyboard scancode FIFO and produce a controller state
Assets	Read named packed data through the pack table or File I/O device
Saves	Maintain RAM shadows and write whole records through File I/O
Graphics	Translate game drawing commands into Voodoo work
Audio	Map live notes and samples onto IE voices

The game code no longer reaches into the old platform. It reaches the contract. The contract reaches Intuition Engine.

## 57.2 Staged Cleanup

---

The porting order matters. A useful sequence is:

1. Make the original game code call named platform functions.
2. Remove platform assumptions from the shared core.
3. Add small IE adapters for one service at a time.
4. Fail explicitly when a service is not present.
5. Add smoke checks for each completed service.

That is slower than replacing everything at once, but it gives each part a testable boundary.

## 57.3 What Stays Out Of The Core

---

The game core should not know:

- Where the keyboard FIFO lives.
- Whether asset bytes came from a packed image or a file device.
- Which Voodoo registers draw a triangle.
- Which IE audio voice plays a sample.

- Where save files are stored.

Those are machine decisions. Keeping them outside the core makes it possible to test the game rules and the IE machine layer separately.

## 57.4 The General IE Lesson

---

For a large IE programme, draw a hard line between behaviour and machine service. The behaviour says what must happen. The IE service layer says which bus address, chip, CPU, or file device makes it happen.