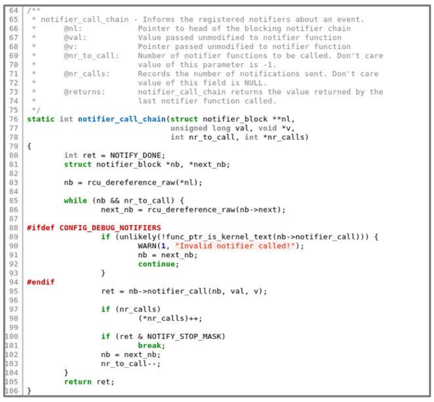
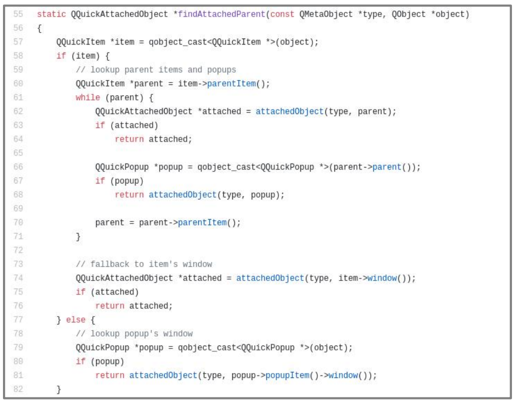
Mobile Secure Computing device

Kernel source code



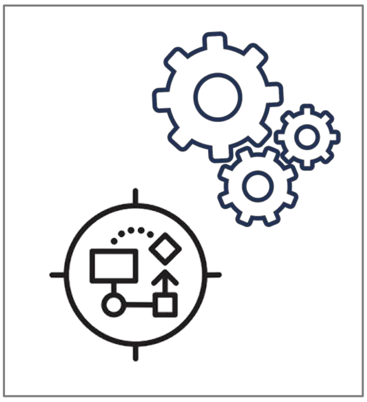
```
/**
 * notifier_call_chain - Informs the registered notifiers about an event.
 *      @nl:            Pointer to head of the blocking notifier chain
 *      @val:           Value passed unmodified to notifier function
 *      @v:             Pointer passed unmodified to notifier function
 *      @nr_to_call:    Number of notifier functions to be called. Don't care
 *                      value of this parameter is -1.
 *      @nr_calls:      Records the number of notifications sent. Don't care
 *                      value of this field is NULL.
 *      @returns:       notifier_call_chain returns the value returned by the
 *                      last notifier function called.
 */
static int notifier_call_chain(struct notifier_block **nl,
                               unsigned long val, void *v,
                               int nr_to_call, int *nr_calls)
{
        int ret = NOTIFY_DONE;
        struct notifier_block *nb, *next_nb;

        nb = rcu_dereference_raw(*nl);

        while (nb && nr_to_call) {
                next_nb = rcu_dereference_raw(nb->next);

#ifdef CONFIG_DEBUG_NOTIFIERS
                if (unlikely(!func_ptr_is_kernel_text(nb->notifier_call))) {
                        WARN(1, "Invalid notifier called!");
                        nb = next_nb;
                        continue;
                }
#endif
                ret = nb->notifier_call(nb, val, v);

                if (nr_calls)
                        (*nr_calls)++;

                if (ret & NOTIFY_STOP_MASK)
                        break;
                nb = next_nb;
                nr_to_call--;
        }
        return ret;
}
```

Applications source code



```
static QQuickAttachedObject *findAttachedParent(const QMetaObject *type, QObject *object)
{
    QQuickItem *item = qobject_cast<QQuickItem *>(object);
    if (item) {
        // lookup parent items and popups
        QQuickItem *parent = item->parentItem();
        while (parent) {
            QQuickAttachedObject *attached = attachedObject(type, parent);
            if (attached)
                return attached;

            QQuickPopup *popup = qobject_cast<QQuickPopup *>(parent->parent());
            if (popup)
                return attachedObject(type, popup);

            parent = parent->parentItem();
        }

        // fallback to item's window
        QQuickAttachedObject *attached = attachedObject(type, item->window());
        if (attached)
            return attached;
    } else {
        // lookup popup's window
        QQuickPopup *popup = qobject_cast<QQuickPopup *>(object);
        if (popup)
            return attachedObject(type, popup->popupItem()->window());
    }
```

Build environment



PriveOS platform

Mobile device

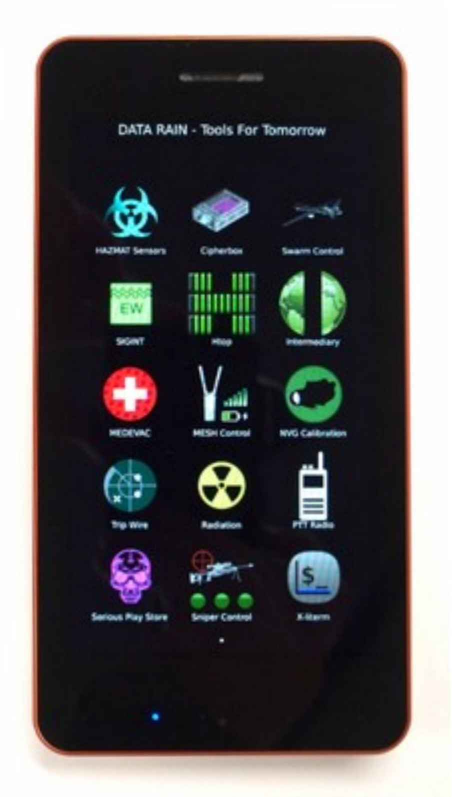
- Mobile Secure Computing device
- Computational freedom
- *Not a mobile phone*

Features

- 100 % Linux computer
- 100 % source visible
- 5.5” Touch Screen
- iMX6 ARM CPU
- 1 GB RAM, 8 GB eMMC
- Audio
- Micro USB (charging only)
- System connector
- 100 Mbit/s hardware Ethernet

Software

- Latest Linux 5.4 kernel
- PriveOS operating system
- This is not Android device
- No ‘known unknowns’ included



Application areas

- Critical Infrastructure IoT
- Red Team exfil tooling
- Authentication platform
- Penetration Testing tool
- Critical Comm’s
- COMSEC & INFOSEC
- MESH network device
- Off The Grid applications

MPP protocol

- Multi Party Consensus protocol
- Authentication, integrity & privacy
- Multiparty keying
- Drop-in ciphers
- Zero META DATA
- Forensically Secure

Software and hardware manufactured in Finland

SUPPLY CHAIN

SOURCE CODE

SCHEMATICS

MECHANICS

MANUFACTURING

yocto PROJECT

GitHub

XXLSEC