# Corso di Programmazione Web e Mobile

# A.A. 2020-2021

# RaspicamNDI web interface

# - Cerutti Davide, 962717 -

# RaspicamNDI

## INTRODUZIONE

RaspicamNDI nasce come esigenza di unire qualità e compattezza, aggiungendo anche una totale indipendenza dalle connessioni cablate, presentando una versione wirless

RaspberryPi è il single board computer più celebre, noto anche per un ottimo rapporto qualità prezzo, l’interfaccia telecamera è in grado di offrire ottime prestazioni e ottime immagini, l’architettura NEON della cpu risc è in grado anch’essa di gestire un discreto numero di processi, tali da consentire lo sviluppo di RaspicamNDI

ideata per la produzione broadcast da Cerutti Davide in collaborazione con VIZRT Group. L’utilizzo della nuova tecnologia NDI 5 (NewTekTM) ha permesso lo sviluppo del flusso NDI su sistemi embedded con CPU risc, rendendo il raspberry un ottimo candidato per la sua applicazione in SDK, implementato in linguaggio C++, ha portato allo sviluppo di RaspicamNDI.

Le esigenze applicative della tecnologia RaspicamNDI hanno richiesto lo sviluppo di una interfaccia web, ponendo le basi di un prodotto, non solo di una tecnologia, di facile utilizzo e implementazione

*RaspicamNDI è sviluppato e ideata da Cerutti Davide;  
NDI è una tecnologia di proprietà di VIZRT Group;  
L’utilizzo della tecnologia NDI 5 è concessa dal Dr. Andrew Cross, president of R&D of Vizrt Group*

La prima implementazione della tecnologia NDI 5 su RaspberryPi è stato un semplice codice, modificabile a sorgente e compilabile, una prima interfaccia a riga di comando è presente dalla versione 1.1.0, contemporaneamente ricevendo l’approvazione dal team di sviluppo NDI. In seguito si è vista l’intenzione di voler trasformare la tecnologia RaspicamNDI in un prodotto utilizzabile con concrete applicazioni nel mondo broadcasting.

L’esigenza primaria è sicuramente stata l’idea di una UI user friendly, non a linea di comando; in oltre, prendendo ispirazione dai moduli di encoding di NewTek e Kiloview, una interfaccia che consenta di gestire l’applicazione in remoto, una interfaccia web.

Da questa idea è stata sviluppata la web interface di raspicamNDI

## DESTINATATRI

RaspicamNDI nasce come tecnologia open source, il codice base è presente come raw-code nel pacchetto SDK NDI5, dunque si rivolge solamente a coloro che intendono sviluppare applicativi alternativi o conoscono i linguaggi di programmazione, una fetta molto ristretta; Mediante aggiunta di UI, web interface e ottimizzazione dei processi RaspicamNDI si trasforma in un prodotto, anche commercializzabile, che si rivolge agli utenti finali

L’interfaccia web di raspicamNDI ha come target tutti coloro che vogliono implementare una nuova tecnologia nella loro infrastruttura di rete NDI, dunque tende a rivolgersi a coloro che già conoscono il prodotto e le meccaniche NDI, ampia conoscenza del mondo della rete e dei sistemi broadcasting, per questo l’interfaccia ripropone tutti gli elementi di routung presentati dall’ “Access Manager”, applicazione di NDI Tools (NewTekTM), conosciuta da tutti gli utilizzatori di NDI.  
Anche dal punto di vista del mondo broadcasting il linguaggio con cui è presentata l’UI è tipica del mondo “Cameraman” con termini familiari come “Esposizione, ISO, Shutter; ecc…)

## MODELLO DI VALORE E DISTRIBUZIONE

RaspicamNDI vista come tecnologia ha un grande valore di ricerca, soprattutto per la consistente implementazione possibile di NDI in sistemi embedded, cambiando anche possibilmente il mondo dei converter di NewTek e Kiloview con i sistemi risc con set di istruzioni NEON

Mediante interfaccia web RaspicamNDI diventa un prodotto, che si posiziona in più ambiti, tra NDI Tools, gratuiti per la distribuzione, NDI SDK e microcamere (in SDI) con costi anche di centinaia di dollari; Per poterne identificarne un valore occorre individuare l’ambito di applicazione nel mondo competitivo del broadcasting, occorre sottolineare come NDI sia notevolmente meno utilizzato di SDI, protocollo con il quale sono scostruite la maggior parte (forse più del 90%) dei service broadcasting, dunque la diffusione di RaspicamNDI come prodotto è commisurata e limitata dal diffondersi della tecnologia NDI.

## FLUSSO DEI DATI

Il vero valore dell’interfaccia web è la sua capacità di generazione dei dati e la possibilità di sincronizzazione di essi a lato server.

Lo storage dei contenuti deve essere eseguito tutto a lato server, l’utilizzo di cookie è incorretto e controproducente.

L’interfaccia prevede una prima selezione di dati da passare in esecuzione e la successiva possibilità di attivare e disattivare l’encoding NDI del buffer di camera, è dunque necessario che ogni device che si connetta in controllo remoto debba visualizzare correttamente lo stato corrente della camera;  
in oltre in caso di spegnimento del dispositivo è utile che i dati possano essere risincronizzati all’accensione con i dati dell’ultima esecuzione, non andando a perdere i settaggi ad ogni spegnimento.

È stato necessario includere una parte di Access Manager per il routing nella applicazione RaspicamNDI per poter regolare i flussi NDI del RaspberryPi (normalmente i software come StudioMonitor, TestPattern, VirtualInput, ecc…, non hanno una parte dei Access Manager propria, ma si appoggiano ad un’altra applicazione “Access Manager” che gestisce il routing dell’intero device al quale i le applicazioni faranno riferimento)

Va in oltre considerato che l’applicativo RaspicamNDI è costruito in linguaggio C++ per poter dare il massimo controllo del flusso di esecuzione per una massima efficienza, dunque è necessario mantenere la medesima linea di pensiero anche per la sua web interface, andando a lavorare sul minor numero di dati possibili ed eseguendone lo storage nel modo più efficiente possibile, eseguendo anche accessi al file JSON ndiconfig, file di interscambio dei dati di Routing del device NDI con le librerie dell’SDK.

## ASPETTI TECNOLOGICI

RaspicamNDI è una interfaccia WEB, HTML5 si è sviluppato nella direzione di un linguaggio per la descrizione delle interfacce rendendolo perfetto per questo scopo.

Seguendo la filosofia strutturale AJAX viene utilizzato HTML5 e CSS3 per la presentazione dei contenuti che vanno a costituire una struttura DOM manipolata mediante lo scripting JavaScript

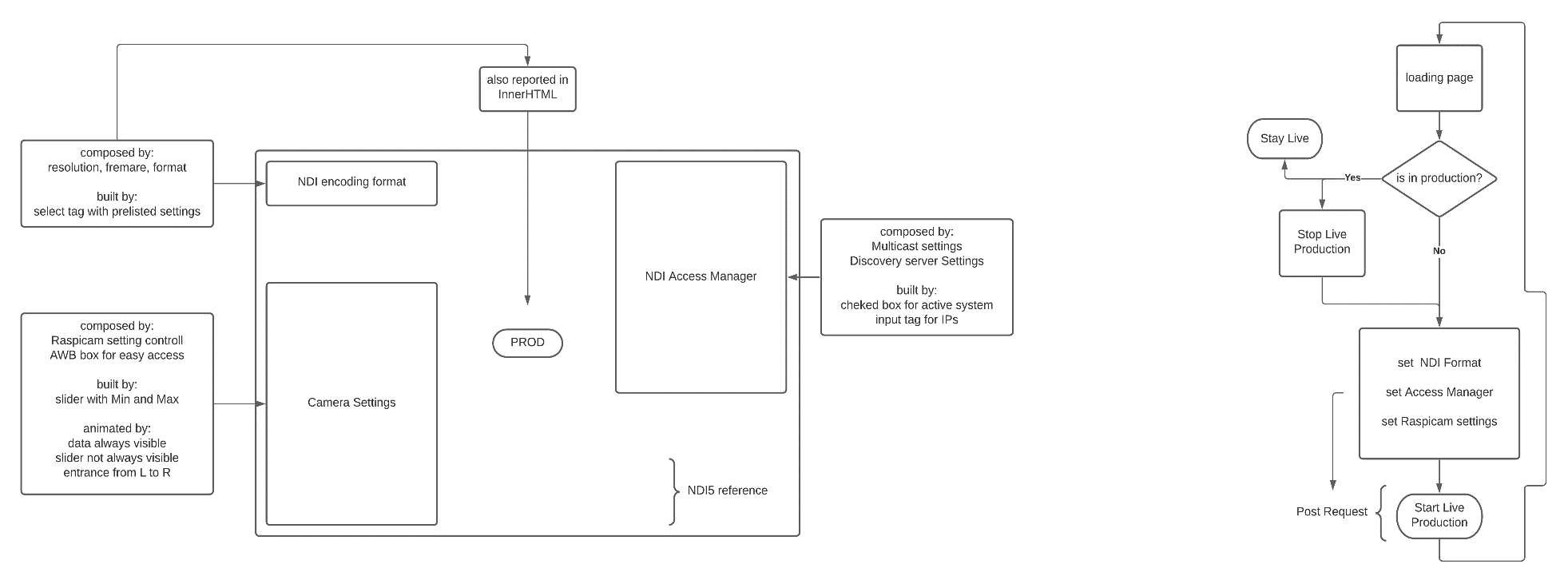
L’interscambio dei dati tra server e browser è regolato però non da XMLHttpRequest ma da placeholder messi a disposizione direttamente dal modulo express EJS, questa scelta è stata subordinata dal fatto che tutti i dati trattati derivano da file JSON, manipolati anche a lato server, e di grande semplicità, i quali non devono essere assolutamente manipolati a lato client, e dunque neanche accessibili da funzioni JS

A lato server viene utilizzato NodeJS, il web server viene creato mediante il modulo EXPRESS, mentre il framework di render è il modulo EJS, utilizzato anche per l’interscambio dati mediante placeholder ( <=% … %> ); altri sono i moduli richiesti da Node a lato server:

* Body-Parser: per interpretare il contenuto del corpo della Req mediante metodo POST
* FS: per accedere ai file di sistema in lettura e scrittura e dunque salvare e caricare dati
* Child-Proces: ExecFile per eseguire l’applicativo exe RaspicamNDI a lato server

## INTERFACCE

RaspicamNDI va a riferirsi ad utenti che conoscono il mondo dell’NDI e lo utilizzano come infrastruttura di trasporto dei segnali video; basando l’interfaccia su questo aspetto si vanno a individuare quelli che sono i classici elementi del “linguaggio broadcasting” e NDI



Come elemento principale vi è il video setting, il formato dello streaming NDI con risoluzione, framerate e format-type, mediante una selezione vengono messi a disposizione le risoluzioni 480, 720 e 1080, i framerate con 60, 50, 30, 25 frame al secondo ( PAL & NTSC ), i format-type progressivo e interlacciato.  
in RaspicamNDI lo spazio colore del frame è fisso al YUV420, che mediante una banda ridotta (simula la vista umana) ottimizza le dimensioni del buffer, notevolmente rispetto ad RGB

Al di sotto del video setting vi sono i controlli della Raspicam, il tipo di input è lo slider, che mette a disposizione un range discreto tra un valore minimo ed un massimo, questi controlli sono i controlli tipici del mondo cameraman e facilmente intuibili, in oltre viene presentato un riquadro per il white balancing relativo ed interattivo ai suoi controlli, nel caso che il bilanciamento del bianco automatico della camera non sia soddisfacente

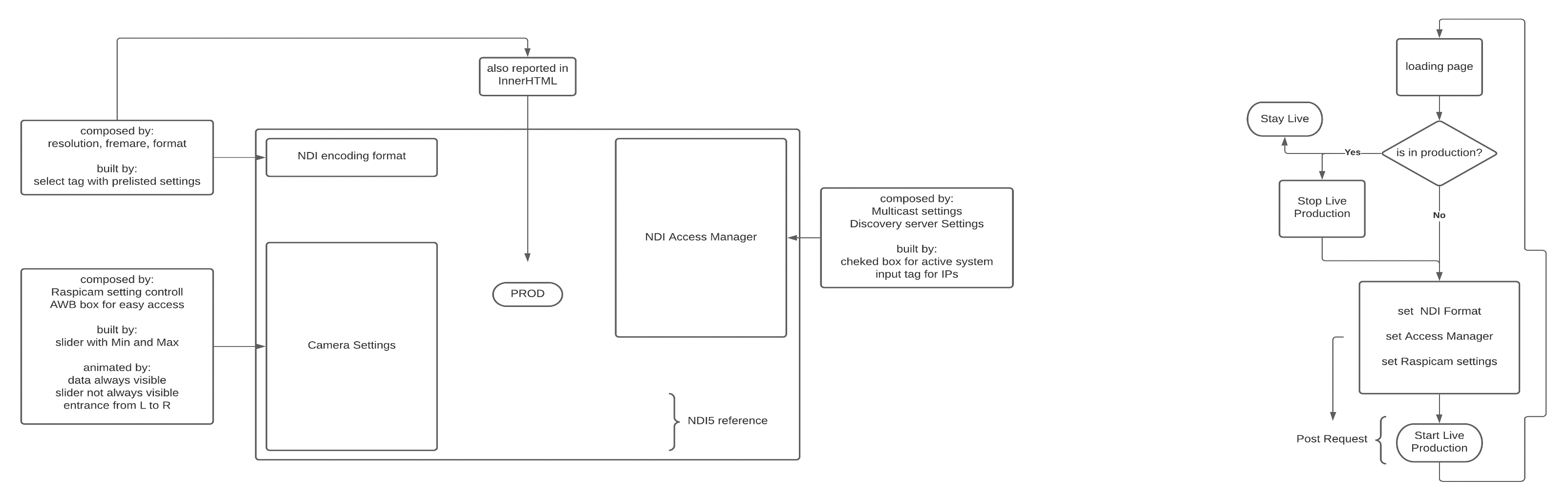
Nella parte sinistra vi è la parte tecninca NDI, la parte di Access Manager, divisa in una sezione relativa al Multicast ed una relativa a Discovery Server; vi è la possibilità di attivare o disattivare queste funzionalità mediante box di checking, i dati sono inseriti mediante form di input, e sono i campi riconducibili direttamente all’Access Manager di NDI NewTek;  
Per riconoscere facilmente l’attivazione o la disattivazione di Discovery e Multicast i compi di input non saranno attivi finché non verrà chekkata la relativa applicazione

Al centro vi è l’ultimo elemento, il button “start live production” è il trigger del metodo POST per l’intero form, che comprende tutti i controlli della pagina e modifica lo steato della pagina stessa e dell’applicativo RaspicamNDI;  
poco sopra vi è un elemento relativo ed indispensabile al “vocabolario broadcasting”, il Tally, la “luce rossa” nel caso di live production, in questo caso “accesa” nel caso si stia producendo il segnale NDI.  
mediante Studio Monitor e l’interpretazione dei metadati del flusso NDI si può vedere il reale Tally di RaspicamNDI durante l’utilizzo come program (non vi è richiesta di proxy resolution in caso di source NDI generata solo come prew e non come program)

Immagine che contiene testo

Descrizione generata automaticamente

La tabella colori di RaspicamNDI web interface si riferisce chiaramente al mondo VIZRT  
L’unico riferimento all’applicativo ed alla sua tecnologia è riportata sulla destra con un rimando alla nuova tecnologia NDI5 che rende possibile la creazione di RaspicamNDI



Il flusso di lavoro è facilmente leggibile e si riferisce soprattutto alla parte sinistra dell’applicazione,   
la parte di Access Manager è da intendersi come un setting “quasi statico”, variabile solo al variare dell’intera struttura NDI del network

Mentre tutti i dati relativi alla camera sono facilmente accessibili ed individuabili nella pagina

Tutti i setting vanno fatti mentre RaspicamNDI non è in stato di produzione!

## ARCHITETTURA DELLE RISORSE

Le risorse utilizzate da RaspicamNDI sono tutte gestite a lato server e vengono salvate in apposti file txt e JSON

Ogni risorsa viene sovrascritta ad ogni nuova istanza di una richiesta di produzione, avendo sempre a lato server la copia aggiornata più recente delle impostazioni di RaspicamNDI

Il file CameraSetting.JSON è il contenitore generale dell’istanza corrente, al suo interno è salvato come oggetto JavaScript tutto il corpo, parsato come oggetto, del body della request, è infatti il file che regola l’intero Model View Controll, tutti i dati mostrati al client derivano dalla lettura di CameraSetting.JSON

Vi sono due file aggiuntivi il cui utilizzo è indipendente dal server, non partecipano al MVC e non sono presentati a lato client, derivano dal corpo di CameraSetting.JSON e sono:

* CameraSetting.txt: un file di testo che contiene tutti i dati che devono essere passati all’applicativo in C++, il quale va a leggere il file binario al posto che pare un parse in JSNO
* ndiconfig.JSON: un file oggetto di JavaScript proprietario del pacchetto SDK NDI5, questo file è la raccolta di tutti i settaggi di routung del dispositivo su rete NDI, in NDI Tools di NewTek è l’Access Manager a scrivere su ndiconfig, in questo caso il file viene sovrascritto con i nuovi valori provenienti dal settore Access Manager di RaspicamNDI, i valori in scrittura sono: Discovery Server ( uno o più indirizzi IPs ) Multicast (sendEnable, TTL, IPs, NetMask )   
  Per mantenere intatta la struttura di ndiconfig.JSON, che deve essere fissa e precisa, viene utilizzato un modello, ovvero un file ndiconfigmodel.JSON, con la stessa struttura oggetto di ndiconfig, ma al quale non viene fatto nessun accesso in scrittura. ndiconfigmodel viene letto a lato server, ritrasformato in oggetto JS mediante parsing, e, implementando i valori di istanza di Access Manager, viene poi sovrascritto il file ndiconfig, così da non richiare mai di perdere la corretta struttura del file ndiconfig

A lato server vi è una variabile globale: “Live\_Track”, questa variabile tiene traccia dello stato di RaspicamNDI, presentando poi al client il corretto stato della camera, se in live production (con anche la presenza del Tally), o se non in live production  
Questa variabile può non essere salvata in memoria ma vivere solo nell’heap del server in quanto ad ogni istanza di server corrisponde una nuova production (ad ogni nuova accensione di RaspicamNDI il live track è sicuramente a 0, dunque non è necessario un ulteriore storaging)

## CONCLUSIONI

RaspberryPi mette a disposizione risorse limitate dalle quali estrarre la migliore ottimizzazione possibile, vi sono a disposizione VNC e SSH come protocolli di accesso da remoto, mediante i quali si potrebbe agire sul raspberry e su RaspicamNDI, ma riguardo all’ottimizzazione non possono essere paragonati all’utilizzo di RaspicamNDI mediante la sua web control page.